



# **Staattisen loistehokompensaattorin ohjausjärjestelmän askelvastetestien automatisointi RTDS-testiympäristössä**

Ilkka Salomaa

Opinnäytetyö  
Maaliskuu 2016  
Automaatioteknologia

# TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Automaatioteknologia

SALOMAA, ILKKA:

Staatin loistehokompensaattorin ohjausjärjestelmän askelvastetestien automatisointi  
RTDS-testiympäristössä

Opinnäytetyö 49 sivua, joista liitteitä 5 sivua  
Maaliskuu 2016

---

Tämän tutkintotyön tarkoituksena on kehittää menetelmä, jolla voidaan automatisoida staatin loistehokompensaattorin ohjausjärjestelmän testaamista. Työ on rajattu koskemaan askelvastetestien automatisointia.

Staatin loistehokompensaattori (engl. Static Var Compensator, SVC) koostuu tyristoriohjatuista keloista, tyristorikytkentäisistä kondensaattoriparistoista ja kiinteistä suodatinparistoista. Tyristoriventtiilien liipaisukulmaa muuttamalla mahdollistetaan portaaton loistehon säätö, minkä ansiosta SVC soveltuu hyvin tilanteisiin, joissa kompensoitavan loistehon määrä ei ole vakio. Sähköverkkoon kytkettävää SVC:tä käytetään muun muassa vakauttamaan kytkentäpisteen jännitettä ja lisäämään verkon häiriökestoisuutta.

SVC:n toiminnallisuus testataan kokonaisuudessaan RTDS-testiympäristössä (engl. Real Time Digital Simulator) ennen sähköasemalla tehtäviä testejä suurjännitteen kanssa. RTDS:n RSCAD-ohjelmalla voidaan mallintaa ja simuloida SVC:n toimintaa sähköverkoissa. SVC:n dynaamisen suorituskyvyn todentaminen on merkittävä osa tehdastestejä ja askelvastetestit ovat yksi suurimmista testattavista kokonaisuuksista.

Ohjausjärjestelmän PID-jännitesäädin viritetään kokeellisesti. Tavoitteena on suoriutua määritellyistä askelvastevaatimuksista. Jos yhdessäkin testitapauksessa ei saavuteta vaadittua suorituskkyä, optimoidaan PID-säätimen parametrit uudelleen kunnes haluttu lopputulos on saavutettu. Tämän vuoksi riippuen testitapausten määrästä askelvastetestien tekeminen manuaalisesti on työläs ja yksitoikkoinen prosessi.

Työssä on luotu RSCAD:llä C-kieleen pohjautuvia komentosarjoja, joiden avulla on toteutettu testisekvenssi, joka automaattisesti ajaa vaadittavat testitapaukset askelvastetesteissä. Lisäksi automatisointia varten RTDS:n ja SVC:n ohjausjärjestelmän välille on luotu kommunikointirajapinta käyttäen DNP3-protokollaa (engl. Distributed Network Protocol). Käyttäjän tarvitsee antaa komentosarjalle lähtöarvot ja analysoida tulokset testien päätteeksi.

Ohjausjärjestelmän ja RTDS:n välisen DNP3-kommunikaation sekä RSCAD:n komentosarjojen testaaminen on antanut lupaavia tuloksia kehitetyn menetelmän toimivuudesta. Alustavien tulosten perusteella testaaminen komentosarjojen avulla on huomattavasti manuaalista testaamista tehokkaampaa.

---

Asiasanat: testiautomaatio, askelvaste, SVC, RTDS, DNP3

## ABSTRACT

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Degree Programme in Automation Technology

SALOMAA, ILKKA:

Step Response Test Automation of Static Var Compensator Control System in RTDS Environment

Master's thesis 49 pages, appendices 5 pages  
March 2016

---

The purpose of this master's thesis was to study and develop a method to automate the testing of the control system of a static var compensator. The thesis was limited to examine the automation of the step response test only.

The static var compensator (abbr. SVC) consists of thyristor-controlled reactors, thyristor-switched capacitors and fixed filter capacitor banks. The reactive power of the SVC can be controlled continuously by changing the firing delay of the thyristors. This makes the SVC suitable for compensating the loads where the reactive power is not static. Transmission SVCs are used to regulate the grid voltage and to improve the dynamic stability of the network.

Factory acceptance tests of the SVC control system are conducted in a RTDS (abbr. Real Time Digital Simulator) test environment. The operation of the SVC in an electrical grid can be modeled and simulated with the RSCAD software of RTDS before the site acceptance tests with high voltage. The verification of the dynamic performance of the SVC is an important part of testing, and the step response tests are one of the biggest test entities.

The PID controller in the SVC control system is tuned experimentally. The aim is to fulfill the customer specified step response requirements. If sufficient performance is not reached in any of the test cases, the PID controller parameters need to be adjusted and the tests have to be started over. Depending on the number of test cases, carrying out step response tests manually is a laborious and repetitive process.

This thesis have created C-language based scripts in RSCAD in order to implement a test sequence that automatically runs all required step response test cases. In addition, a communication interface between the RTDS and the SVC control system has been implemented by using the DNP3 protocol (abbr. of Distributed Network Protocol). Due to the automation the user only needs to give the input data to the script and to analyze the results after the test sequence has been finished.

The testing of the DNP3 communication and the RSCAD test sequence scripts has provided promising results of the functionality of the developed method. According to the initial results, testing is considerably more efficient by using the automated test sequence than testing manually.

---

Key words: test automation, step response, SVC, RTDS, DNP3

## SISÄLLYS

|       |  |    |
|-------|--|----|
| 1     | JOHDANTO.....                                | 7  |
| 2     | TYÖHÖN LIITTYVÄN TEKNIIKAN ESITTELY .....    | 8  |
| 2.1   | Staattinen loistehokompensaattori, SVC ..... | 8  |
| 2.2   | NDC-ohjausjärjestelmä.....                   | 12 |
| 2.2.1 | Ohjausjärjestelmän rakenne .....             | 12 |
| 2.2.2 | Säätötavat .....                             | 13 |
| 2.3   | RTDS-simulointiympäristö.....                | 16 |
| 2.4   | DNP3-protokolla.....                         | 19 |
| 3     | SÄÄTÖJÄRJESTELMÄN VIRITTÄMINEN.....          | 23 |
| 3.1   | PID-algoritmi .....                          | 23 |
| 3.2   | Askelvastetesti .....                        | 25 |
| 3.3   | Säätimen virittäminen .....                  | 27 |
| 4     | TESTIAUTOMAATIO .....                        | 29 |
| 4.1   | Perusteet.....                               | 29 |
| 4.2   | Testiautomaation työkaluja.....              | 30 |
| 5     | ASKELVASTETESTIEN AUTOMATISOINTI.....        | 32 |
| 5.1   | Nykytilanne.....                             | 32 |
| 5.2   | Automatisoinnin toteutus.....                | 36 |
| 5.2.1 | DNP3-protokollan käyttöönotto.....           | 36 |
| 5.2.2 | RSCAD-komentosarjat .....                    | 38 |
| 5.3   | Automatisoinnin tulokset.....                | 40 |
| 6     | YHTEENVETO .....                             | 42 |
|       | LÄHTEET .....                                | 44 |
|       | LIITTEET .....                               | 45 |
|       | Liite 1. GTNET DNP Device Profile .....      | 45 |

**LYHENTEET JA TERMIT**

|          |  |
|----------|--|
| COMTRADE | Common format for Transient Data Exchange for power systems                                    |
| CMI      | Communication and Maintenance Interface  |
| CRC      | Cyclic Redundancy Check, virheentarkistusalgoritmi   |
| DNP      | Distributed Network Protocol   |
| DNP3     | Kolmannen sukupolven DNP   |
| FACTS    | Flexible AC Transmission System  |
| GTAI     | Giga-Transceiver Analog Input  |
| GTAO     | Giga-Transceiver Analog Output   |
| GTDI     | Giga-Transceiver Digital Input   |
| GTDO     | Giga-Transceiver Digital Output  |
| HMI      | Human–Machine Interface  |
| I/O      | Input/Output   |
| IEC      | International Electrotechnical Commission, kansainvälinen sähköalan standardointiorganisaatio, |
| IED      | Intelligent Electronic Device  |
| NDC      | Nokian Digital Control, digitaalinen ohjausjärjestelmä   |
| RTDS     | Real Time Digital Simulator, reaaliaikainen digitaalisimulaattori                              |
| RTU      | Remote Terminal Unit   |
| SCL      | Short Circuit Level, sähköverkon oikosulkuteho   |
| OLE      | Object Linking and Embedding   |
| OPC      | OLE for Process Control, avoimen tiedonsiirron standardi                                       |
| OSI      | Open Systems Interconnection Reference Model   |
| PID      | Proportional-Integral-Derivative   |
| PCC      | Point of common coupling, sähköverkon liityntäpiste  |
| PLC      | Programmable Logic Controller  |
| POD      | Power Oscillation Damping  |
| pu       | Per unit, nimellispisteeseen suhteutettu arvo  |
| RS-232   | Recommended Standard 232, sarjaliikennestandardi   |
| RS-485   | Recommended Standard 485, sarjaliikennestandardi   |
| STATCOM  | Static Compensator, staattinen synkronikompensaattori.   |
| SVC      | Static Var Compensator, staattinen loistehokompensaattori                                      |

|        |  |
|--------|--|
| TCP/IP | Transmission Control Protocol / Internet Protocol                                  |
| TCSC   | Thyristor Controller Series Capacitor, tyristoriohjattu sarjakondensaattoriparisto |
| TCR    | Thyristor Controlled Reactor, tyristoriohjattu kela                                |
| TSC    | Thyristor Switched Capacitor, tyristorikytketty kondensaattori                     |
| SCADA  | Supervisory Control And Data Acquisition   |
| VBA    | Visual Basic for Applications  |
| VME    | Versa Module Eurocard  |

## 1 JOHDANTO

GE Grid Solutions Oy:ssä staattisen loistehokompensaattorin (engl. Static Var Compensator, SVC) ohjausjärjestelmän testaamisessa käytetään RTDS-testiympäristöä (engl. Real Time Digital Simulator). Lopputyön tarkoituksena on kehittää menetelmä automatisoida ohjausjärjestelmän testaamista.

Työssä keskitytään ohjausjärjestelmän askelvastetesteihin, joka on yksi työläimmistä testikokonaisuuksista, sillä erilaisia testitapauksia voi olla kymmenittäin. SVC:n tulee kyetä vastaamaan vasteaikavaatimuksiin erilaisissa sähköverkon ja SVC:n toimintapisteissä. Jos yhdessäkin tilanteessa vasteaikavaatimukset eivät täyty, niin vaihdetaan ohjausjärjestelmän PID-säätimen parametreja ja ajetaan kaikki testitapaukset uudelleen, kunnes saavutetaan vaatimukset.

Työssä luodaan RTDS:ään C-kielisiä komentosarjoja, joita hyödynnetään askelvastetestien automatisoinnissa. Lisäksi RTDS:n ja ohjausjärjestelmän välille luodaan kommunikaatorajapinta käyttäen DNP3-protokollaa.

Kehitetty menetelmä pyrkii minimoimaan mekaaniseen toistoon käytettävän ajan, jolloin automaatio hoitaa esim. parametrien asettelut, askelvasteet ja tulosten tallentamisen. Testaajan tarvitsee antaa vain lähtöarvot automaatiolle ja analysoida tulokset testien päätteeksi. Testiprosessia tehostamalla mahdollistetaan yrityksessä tehokkaampi resursointi, kun käytettävissä olevaa työaikaa saadaan lisättyä muihin työtehtäviin.

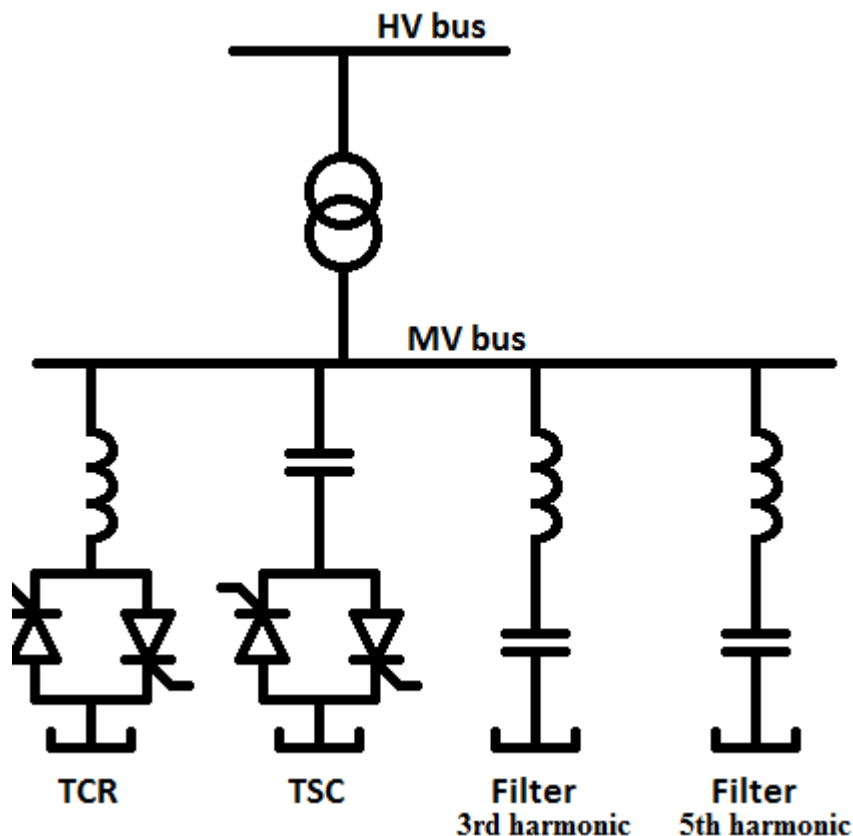
Luvussa 2 esitellään staattisen loistehokompensaattorin toimintaa ja komponentteja sekä SVC:n ohjausjärjestelmän rakennetta ja säätötapoja. Lisäksi luvussa käsitellään RTDS-testiympäristöä ja DNP3-kommunikointiprotokollaa. Luvussa 3 tarkastellaan dynaamista säätöjärjestelmää ja siihen liittyviä lainalaisuuksia analogisen säädön, PID-säätimen virittämisen ja askelvasteen kautta. Luvussa 4 käydään läpi testiautomaatioon liittyvät peruskäsitteet. Luvussa 5 tehdään katsaus yrityksen tämän hetkiseen tapaan tehdä askelvastetestejä ja kuvataan kuinka testien automatisointi voidaan toteuttaa.

## 2 TYÖHÖN LIITTYVÄN TEKNIIKAN ESITTELY

### 2.1 Staattinen loistehokompensaattori, SVC

Teollisuus- ja sähköverkkoympäristöissä käytettävä staattinen loistehokompensaattori eli SVC kuuluu FACTS-laitteisiin (engl. Flexible Alternating Current Transmission System), joilla saadaan kompensoitua loistehoa ja sen myötä parannettua verkon sähkön laatua. Muita FACTS-laitteita ovat esimerkiksi STATCOM ja TCSC.

SVC:n päämuuntajan yläjännitepuoli kytketään kantaverkkoon PCC-liityntäpisteeseen (engl. Point of Common Coupling) ja SVC:n pääkomponentit on kytkettynä päämuuntajan alajännitepuolella keskijännitekiskostossa, jossa jännite on tyypillisesti 10...36 kV:n välillä. Kuviossa 1 on esitettyä tyypillisen SVC:n pääkomponentit yksiviivakaaviossa.

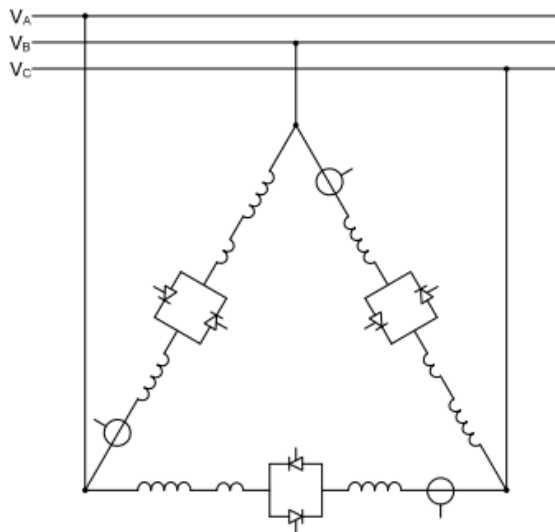


KUVIO 1. Tyypillisen SVC:n yksiviivakaavio (Kundur 1994, 653)

Tyristoriorhjattu kela eli TCR (Thyristor Controller Reactor) koostuu kolmioon kytketyistä reaktoreista, joiden virtaa voidaan säätää jatkuvasti ajon aikana

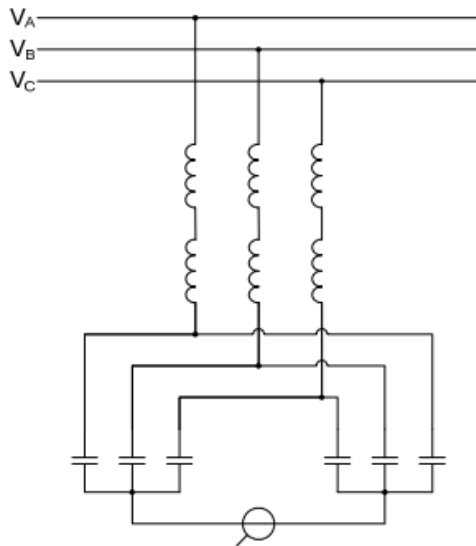


tyristoriventtiileillä. Kytkeä on esitettyä kuviossa 2. Lisäksi kytkennässä on vaimennuskela, joka vaimentaa nopeita hajakapasitansseista purkautuvia virtapiikkejä ja suojaa tyristoriventtiiliä. Tyristoriventtiilien avulla voidaan säätää reaktoreiden tuottamaa induktiivista loistehoa. Tyristoriventtiili muodostuu kahdesta sarjaankytkettyjen tyristorien muodostamasta haarasta, yksi kumpaakin puolialtoa varten. Tyristoriventtiilin haarat ovat rinnankytketyt toisiinsa jokaisella tyristoritasolla, joten vastarinnankytkettyjen tyristoriparien sarja muodostaa venttiilin. (Nokian Capacitors 2009.)



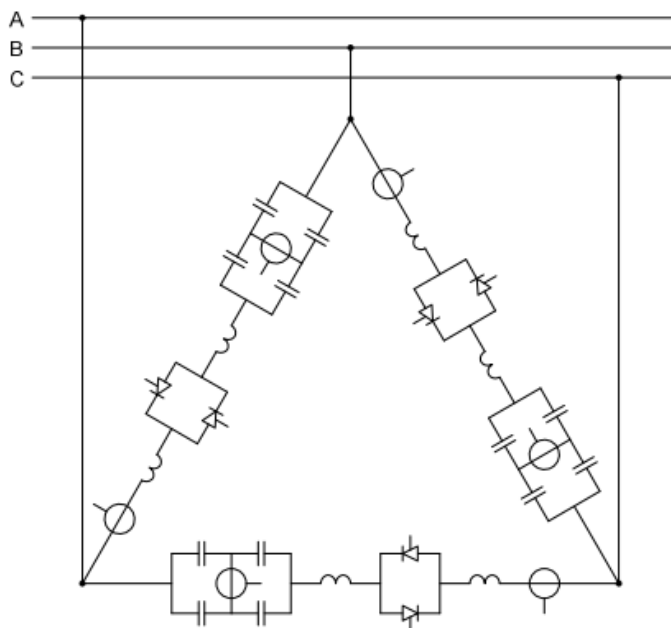
KUVIO 2. TCR-haaran kytkentä (Nokian Capacitors 2009)

Suodatinparistot koostuvat kaksoistähtikytkentäisestä kondensaattoriparistosta ja sen kanssa sarjassa olevasta reaktorista. Kytkeä on esitettyä kuviossa 3. Kaksoistähtikytkentä mahdollistaa suodatinpariston epäbalanssisuojauksen. Kondensaattoriparistojen ja reaktoreiden arvot ovat valittu siten, että ne toimivat imupiireinä halutun kertaluvun harmonisille virroille. Tällä tavoin suodatinparistot vaimentavat verkossa olemassa olevaa ja TCR:n tuottamaa säröä. Toinen käyttötarkoitus suodatinparistoille on tarvittavan kapasitiivisen loistehon tuottaminen. (Nokian Capacitors 2009.)



KUVIO 3. Kaksoistähteen kytketty suodatinparisto (Nokian Capacitors 2009)

Tyristorikytkeetyt kondensaattoriparistot eli TSC:t (Thyristor Switched Capacitor) ovat kapasitiivisia komponentteja, jotka koostuvat kolmioon kytketyistä TSC-haaroista, jotka sisältävät H-kytkentäisen kondensaattoripariston, tyristoriventtiilin ja vaimennusreaktorin. Kytkentä on esitettyinä kuviossa 4. TSC:n tarkoitus on laajentaa SVC:n kapasitiivista toiminta-aluetta ja se kytketään päälle kun tarve kapasitiiviselle loisteholle on enemmän kuin mitä suodatinparistot pystyvät tuottamaan. TSC:n virtaa ei voida säätää portaattomasti kuten TCR:llä, vaan se joko johtaa täysin tai ei ollenkaan. (Nokian Capacitors 2009.)



KUVIO 4. TSC-haaran kytkentä (Nokian Capacitors 2009)

Kuvassa 1 on esitettyä SVC:n ulkoasenteisia pääkomponentteja sähköasemalla. Kuvassa näkyy alalaidassa TSC-kondensaattoriparistot, keskellä TCR-kelat ja niiden takana kolme suodatinparistoa. Tyristoriventtiilit ja SVC:n ohjaus- ja suojauskaapit on sijoitettu valvomorakennuksen sisälle.



KUVA 1. SVC:n pääkomponentteja sähköasemalla (Kangasala SVC 2008)

TCR mahdollistaa erittäin nopean ja portaattoman loistehon säädön muuttamalla tyristorien liipaisukulmaa. Juuri portaattoman säädön vuoksi SVC soveltuu hyvin tilanteisiin, joissa kompensoitavan loistehon määrä ei ole vakio.

Teollisuudessa SVC:tä käytetään kompensoimaan terästehtaiden valokaari- ja senkkauunien sekä valssilaitosten aiheuttamaa nopeasti tapahtuvaa loistehon ja jännitteen muutosta. Sähköverkkopuolella SVC:tä käytetään vakauttamaan ja säätämään kytkentäpisteen jännitettä, poistamaan jännitteiden epäsymmetriaa, nostamaan linjan siirtokapasiteettia, lisäämään verkon häiriökestoisuutta ja käyttövarmuutta sekä parantamaan kantaverkon tehoheilahtelujen vaimentamista. (Kundur 1994, 654.)

## 2.2 NDC-ohjausjärjestelmä

### 2.2.1 Ohjausjärjestelmän rakenne

GE Grid Solutions Oy:ssä on käytössä alun perin Nokian Capacitors Oy:n kehittämä NDC-säätöjärjestelmä (Nokian Digital Control) SVC:n ohjaamista varten. Ohjauslaitteiston tarkoitus on säätää SVC:n tuottamaa loistehoa sen mukaan, mikä säätötapa on valittuna. Ohjausjärjestelmä kerää tietoja itsestään, suojausjärjestelmästä, venttiiliohjauskaapeilta, jäähdytysjärjestelmiltä, apusähköjärjestelmiltä, katkaisijoilta ja erottimilta. Ohjausjärjestelmä tekee logiikkaohjelmallaan päättelyt kerätyistä tiedoista ja ohjaa tarpeen mukaan tyristoriventtiilejä, erottimia, katkaisijoita ja tyristoriventtiilien jäähdytysjärjestelmiä. (Nokian Capacitors 2009.)

NDC-ohjausjärjestelmä pohjautuu VME-väylään, johon on liitetty tyypillisesti kolme prosessorikorttia (Master, Slave ja CMI), joissa PLC-ohjelmaa ajetaan. PLC-ohjelmat on toteutettu NDC-ohjelmointikielellä ja säätölogiikkaa suoritetaan diskreetisti 100 µs, 1 ms, 10 ms tai 100 ms:n aikatasoissa. Prosessorina toimii Motorolan MVME4100-mikroprosessoriperheen MPC8548E 1,3 GHz:n mikroprosessori. Kyseistä prosessorityyppiä ja VME-väylää käytetään yleisesti teollisuudessa ja asevoimien sovelluksissa. (Nokian Capacitors 2009.)

Master-prosessorikortin PLC-ohjelmassa käsitellään SVC:n loistehon ohjaus analogisiin mittauksiin ja järjestelmän toimintatilaan perustuen. Slave-prosessorikortin PLC-ohjelma vastaa kentältä tulevien tilatietojen käsittelystä ja apulaitteiden ohjauksesta. CMI-prosessorikortti (Communication & Maintenance Interface) toimii VME:n kommunikointialustana ja mahdollistaa OPC-rajapinnan HMI:n kanssa ja kommunikoinnin ulkoisiin sähköaseman SCADA-järjestelmiin. Yleisimmin käytettyjä kommunikointiprotokollia ovat DNP3, IEC 60870-5-104 ja IEC 61850.

Säätöpiiriin saadaan mittaukset suurjännitekiskostoon kytketyiltä jännite- ja virtamuuntajilta. Lisäksi keskijännitekiskostoon kytketyltä jännitemuuntajalta saadaan jännitemittaus, jota käytetään TCR-liipaisupulssien synkronisointia varten. Ohjausjärjestelmä lähettää tyristoriohaukset valokuiduilla tyristoriventtiileille. Venttiilielektroniikka muuttaa valopulssit sähköisiksi tyristorien ohjauspulsseiksi.

Vastaavasti tyristoriventtiileiltä saadaan valokuitua pitkin takaisin diagnostiikkatietoa mahdollisista vioista tai häilytyksistä. (Nokian Capacitors 2009.)

### 2.2.2 Sääötötavat

SVC:n tuottama loisteho riippuu TCR:n liipaisukulmasta, päämuuntajan läpi menevästä virrasta, TSC:iden ja suodatinparistojen tuottamasta kapasitiivisesta loistehosta ja jännitetasosta ylä- ja alajännitepuolella. SVC:n säätöalueen ajatellaan siis olevan koko SVC:n susceptanssi  $B_{SVC}$  kaavan 1 mukaisesti.

$$B_{SVC} = \frac{(B_{suod} + B_{TSC} + B_{TCR}) \cdot B_m}{B_{suod} + B_{TSC} + B_{TCR} + B_m}, \quad (1)$$

missä  $B_{suod}$  on suodatinparistojen,  $B_{TSC}$  TSC:n,  $B_{TCR}$  TCR:n ja  $B_m$  päämuuntajan susceptanssi.

SVC:n susceptanssia ja sen tuottamaa loistehoa ohjataan muuttamalla TCR:n liipaisukulmaa ja ohjaamalla TSC:n kytkentätilaa. Ohjausjärjestelmän PLC-logiikka laskee tehon tarpeen uudelleen 100 mikrosekunnin välein. Tarvittava teho TCR:n ohjausta varten muunnetaan PLC-koodissa tyristorin liipaisuviiveeksi. Susceptanssiohje ei suoraan käy tyristoriventtiilien liipaisuohjeeksi, vaan susceptanssiohje linearisoidaan toimilohkon avulla, jotta saadaan laskettua susceptanssia vastaava TCR:n liipaisukulma. (Nokian Capacitors 2009.)

Suurin induktiivinen teho saadaan silloin kun TCR:n liipaisukulmat ovat minimissään ja TSC on kytketty pois päältä. Suurin kapasitiivinen teho saadaan puolestaan kytkemällä TSC päälle ja ajamalla TCR:n liipaisukulmat maksimiinsa. Loistehon ollessa nolla TSC on poiskytkettynä ja TCR:n avulla kompensoidaan filttareiden aiheuttama kapasitiivinen loisteho.

Kaikille sääötötavoille yhteistä on se, että ohjausjärjestelmä reagoi valitun sääötötavan mukaisesti johonkin tulosiinaaliin muuttamalla SVC:n susceptanssia. Susceptanssia ohjataan alueella, jonka ääriarvot määräytyvät SVC-kiskoon kytkettyjen reaktiivisten ja kapasitiivisten komponenttien summana. (Nokian Capacitors 2009.)

Säätimen ulostuloon vaikuttaa myös sähköverkon oikosulkuteho, joka kuvastaa kytkentäpisteen sähköistä jäykkyyttä. Mitä suurempi oikosulkuteho, sen suurempi loistehomuutos tarvitaan muuttamaan jännitteen arvoa. Vastaavasti pienemmällä oikosulkuteholla riittää pienempi loistehon muutos saman jännitemuutoksen aikaansaamiseksi. SVC:n tuottama loisteho muuttaa kytkentäpisteen jännitettä  $U_{PCC}$  sähköverkon oikosulkutehoon riippuvaisesti kaavan 2 mukaisesti.

$$\Delta U_{PCC} = \frac{\Delta Q_{SVC}}{S_{SCL}}, \quad (2)$$

missä  $Q_{SVC}$  on SVC:n tuottama loisteho ja  $S_{SCL}$  verkon oikosulkuteho.

Sähköverkko-SVC voi toimia neljällä eri säätötavalla:

- Jännitesäädössä SVC pyrkii kytkemään laitteitaan siten, että suurjännitekiskon jännite pysyy asetusarvossaan.
- Loistehonsäädössä SVC pyrkii kytkemään laitteitaan siten, että loisteho pysyy asetetussa arvossa. Loistehosäädin on hidas säätötapa, mutta toimii kaskadisäätönä jännitesäätimen kanssa, jolloin pystytään nopeasti reagoimaan esimerkiksi vikatilanteisiin.
- POD-säädössä (Power Oscillation Damping) SVC pyrkii vaimentamaan sähköverkossa esiintyviä tehoheilahteluja. POD-säädössä mitataan suurjännitekiskon taajuutta, jonka heilahtelut aiheuttavat SVC:n loistehon muutoksen.
- Manuaalisäädössä pystytään ohjaamaan SVC:n susceptanssia haluttuun arvoon, mutta säädin ei reagoi sähköverkossa tapahtuviin muutoksiin. Manuaalisäätöä käytetään ainoastaan käyttöönotto- tai testaustilanteissa. (Nokian Capacitors 2009.)

### Jännitesäätö

Jännitesäädössä käyttäjä asettaa suurjännitekiskon jännitteen referenssiarvon. Ohjaussovellus pyrkii pitämään kiskon jännitteen asetetussa arvossa muuttamalla SVC:n susceptanssia kiskon jännitemuutosten mukaisesti. Jännitesäätötilassa ohjaussovellus vastaa vasteaikavaatimusten mukaisesti suurjännitekiskossa tapahtuviin jännitemuutoksiin. (Nokian Capacitors 2009.)

Jännitesäätö on toteutettu PID-säätimellä, jonka sisäänmenona käytetään jännitevirhettä. Jännitevirheen arvon määräytymiseen vaikuttaa suurjännitekiskon jännite, käyttäjän asettama jännitereferenssi, SVC:n tuottama loisteho sekä käyttäjän asettama slope-asetus. PID-säätimen lähdön perusteella ohjataan SVC:n susceptanssia eli SVC:n tuottamaa loistehoa. (Nokian Capacitors 2009.)

Slope-asetuksen  $X_{\text{slope}}$  avulla käyttäjä voi vaikuttaa siihen, miten suurta SVC:n tehoa ohjaussovellus käyttää suhteessa kisko-jännitteen virheeseen eli miten suurella kulmakertoimella SVC reagoi jännitteen muutoksiin. Kytkeänpisteen jännitteen  $U_{\text{PCC}}$  laskenta on esitetty kaavassa 3.

$$U_{\text{PCC}} = U_{\text{ohje}} - I_{\text{PCC}} \cdot X_{\text{slope}}, \quad (3)$$

missä  $U_{\text{ohje}}$  on SVC:n jännitesäätimen ohjearvo,  $I_{\text{PCC}}$  SVC:n virta yläjännitepuolella ja  $X_{\text{slope}}$  slope-asetus prosentteina ilmoitettuna. (Kundur 1994, 643.)

Mitä suurempi slopen arvo on, sitä pienempää SVC:n tehoa käytetään suhteessa jännitevirheen suuruuteen. Slopen arvon ja SVC:n tuottaman loistehon tulona lasketaan droop-arvo, jota käytetään vähentämään PID-säätimelle syötettävää virhettä. Mikäli slopen arvoksi asetetaan nolla ainoastaan mitatun jännitteen ja asetetun jännitereferenssin erotuksella on vaikutusta säätimen toimintaan.

SVC:n tuottamaa loistehoa muutetaan induktiiviseen suuntaan, mikäli suurjännitekiskon jännite nousee asetetun referenssiarvon yläpuolelle. Vastaavasti SVC:n tuottamaa loistehoa muutetaan kapasitiiviseen suuntaan, mikäli kiskon jännite laskee asetetun referenssiarvon alapuolelle. SVC:n tuottama loisteho tilanteen mukaan joko nostaa tai laskee kiskon jännitettä ja näin aiheuttaa säätimen takaisinkytkennän.

### **Loistehonsäätö**

Loistehonsäädössä käyttäjä asettaa SVC:n tuottamalle loisteholle viitearvon. Ohjaussovellus pyrkii pitämään SVC:n tuottaman loistehon asetetussa arvossa muuttamalla SVC:n susceptanssia. Loistehon säätötilassa ohjaussovellus reagoi ainoastaan loistehon muutoksiin, ja vasteaika on hitaampi kuin jännitesäätötilassa. Loistehosäädin toimii kuitenkin kaskadisäätönä jännitesäätimen kanssa, jolloin

pystytään nopeasti reagoimaan esimerkiksi verkossa tapahtuviin vikatilanteisiin. (Nokian Capacitors 2009.)

SVC:n tuottama loisteho lasketaan suurjännitekiskon jännitemittauksen ja SVC-kiskon virtamittauksen perusteella. SVC:n mitatun loistehon ja käyttäjän asettaman loistehoreferenssin erotuksena saadaan loistehovirhe, jota käytetään syötteenä säädön toteuttavalle PID-säätimelle. Säätimen lähdön perusteella ohjataan SVC:n suskeptanssia. PID-säädin pyrkii pitämään loistehovirheen mahdollisimman pienenä. (Nokian Capacitors 2009.)

### **POD-säätö**

POD-säädössä SVC pyrkii vaimentamaan sähköverkossa esiintyviä matalataajuisia tehoheilahteluja. POD-säädössä mitataan suurjännitekiskon taajuutta, jonka heilahtelut aiheuttavat SVC:n loistehon muutoksen. Operaattori valitsee parametrisetin joka kuvaa miten SVC:n suskeptanssi muuttuu suhteessa valitun mittauksen muutoksiin. (Nokian Capacitors 2009.)

### **Manuaalisäätö**

Manuaalisäädössä käyttäjä antaa haluamansa SVC:n suskeptanssin arvon parametrina ohjaussovellukselle. Ohjaussovellus tarkistaa käyttäjän antaman suskeptanssiarvon, jonka jälkeen SVC:n suskeptanssi rampitetaan asetettuun arvoon. Manuaalisäädössä SVC:n ohjaus perustuu pelkästään käyttäjän asettamaan referenssiarvoon eikä järjestelmä reagoi esimerkiksi suurjännitekiskossa tapahtuvaan äkilliseen jännitemuutokseen. (Nokian Capacitors 2009.)

## **2.3 RTDS-simulointiympäristö**

SVC:n ohjausjärjestelmä testataan RTDS Technologies Inc.:n kehittämällä reaaliaikaisella digitaalisimulaattori RTDS:llä (Real Time Digital Simulator). RTDS mahdollistaa sähköverkon mallintamisen ja simuloinnin reaaliajassa, jolloin siihen kytkettyjen laitteiden käyttäytymistä voidaan tutkia ilman, että vaarannetaan todellisen sähkövoimajärjestelmä toimintaa. RTDS soveltuu niin laajojen siirto- ja jakeluverkkojen kuin laivojen ja lentokoneiden sähköjärjestelmien mallintamiseen.



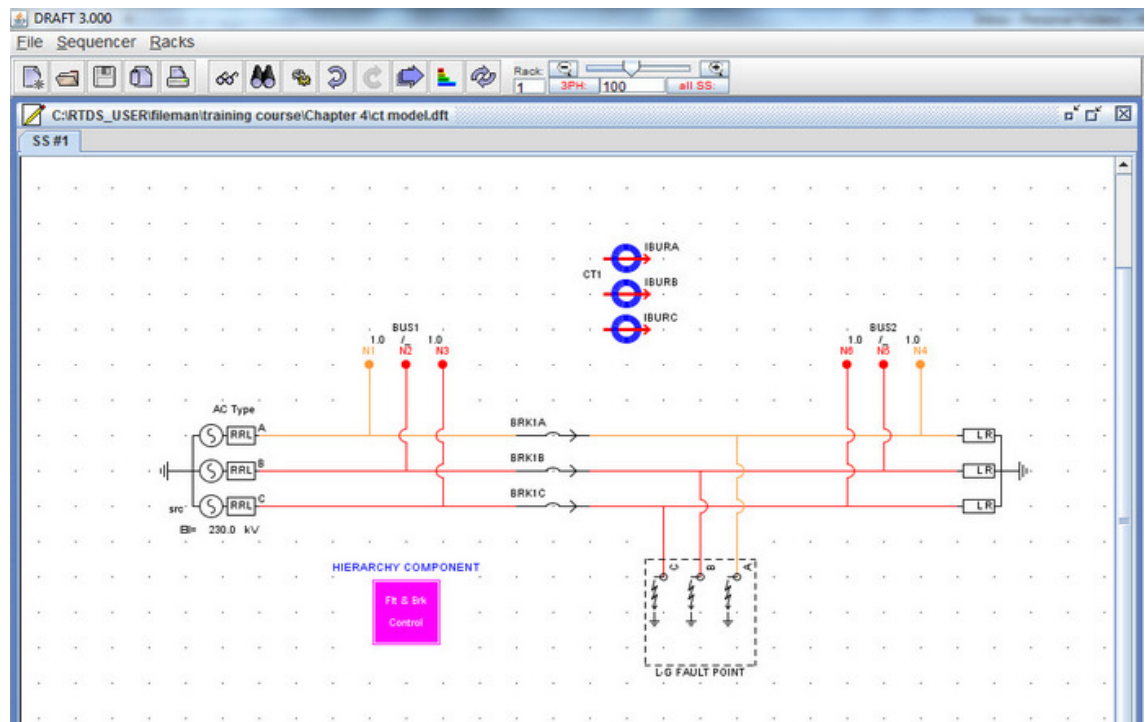
RTDS koostuu räkiasenteisista GPC- tai PB5-mallisista prosessorikorteista, joissa verkkomallin laskenta tapahtuu. Käytössä on myös I/O-moduuleita (GTDI, GTDO, GTAI ja GTAO), joiden avulla saadaan toteutettua digitaaliset ja analogiset tulot ja lähdöt RTDS:n ja testattavan ohjausjärjestelmän välillä. GTWIF- ja WIF-kortit toimivat Ethernet-kaapelin kautta rajapintana RTDS:n ja PC:n välillä. GTNET-kortin avulla saadaan toteutettua kommunikointirajapinta RTDS:n ja testattavana olevan laitteiston välille. GTNET-kortin tuettuja protokollia ovat IEC 61850, IEC 60870-5-104 ja DNP3. Kuvassa 2 on esitettynä RTDS-räkki ja siihen asennettuja piirikortteja. (RTDS 2015).



KUVA 2. RTDS-räkki (RTDS 2015)

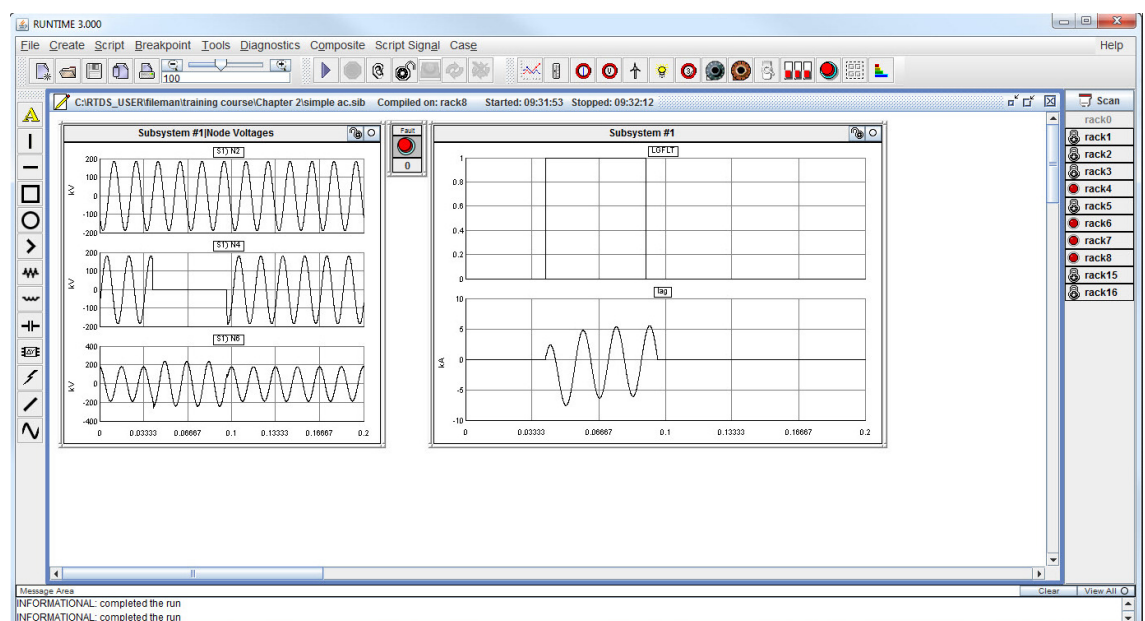
Simuloinneissa käytetään tyypillisesti 50 mikrosekunnin aika-askelta, joka riittää hyvin SVC:n ohjausjärjestelmän testaamiseen. Kytkemällä enemmän prosessorikortteja RTDS:ään voidaan lisätä sen laskentakapasiteettia ja näin ollen kasvattaa verkkomallin laajuutta.

RSCAD on graafinen käyttöliittymä RTDS-simulaattorin käyttämiseen. RSCAD:n Draft-tilaa käytetään simulaatiomallin rakentamiseen ja ohjelman kirjastosta löytyy valmiina kattava kokoelma sähköverkon eri komponentteja. Malli piirretään valitsemalla halutut komponentit kirjastosta ja yhdistämällä ne johdoilla. RSCAD:ssä on myös mahdollisuus rakentaa omia toimilohkoja, jotka kirjoitetaan C-pohjaisella ohjelmointikielellä. Kuvassa 3 on esitettynä näkymä Draft-tilasta. (RTDS 2015.)



KUVA 3. Näkymä RSCAD:n Draft-tilasta

Runtime-tilassa voidaan ohjata ja tarkastella simulointia (kuva 4). Sähköverkon komponenttien parametrejä voidaan muuttaa simuloinnin ollessa päällä, mikä mahdollistaa nopeasti ja helposti esimerkiksi ohjausjärjestelmän toiminnan tarkastelun erilaisissa peräkkäisissä vikatilanteissa. RSCAD:ssä voidaan luoda C-pohjaisella ohjelmointikielellä komentosarjoja, joiden avulla voidaan asettaa esimerkiksi laajan sähköverkkomallin kaikki kytkinlaitteet haluttuun asentoon yhdellä napin painalluksella. (RTDS 2015.)



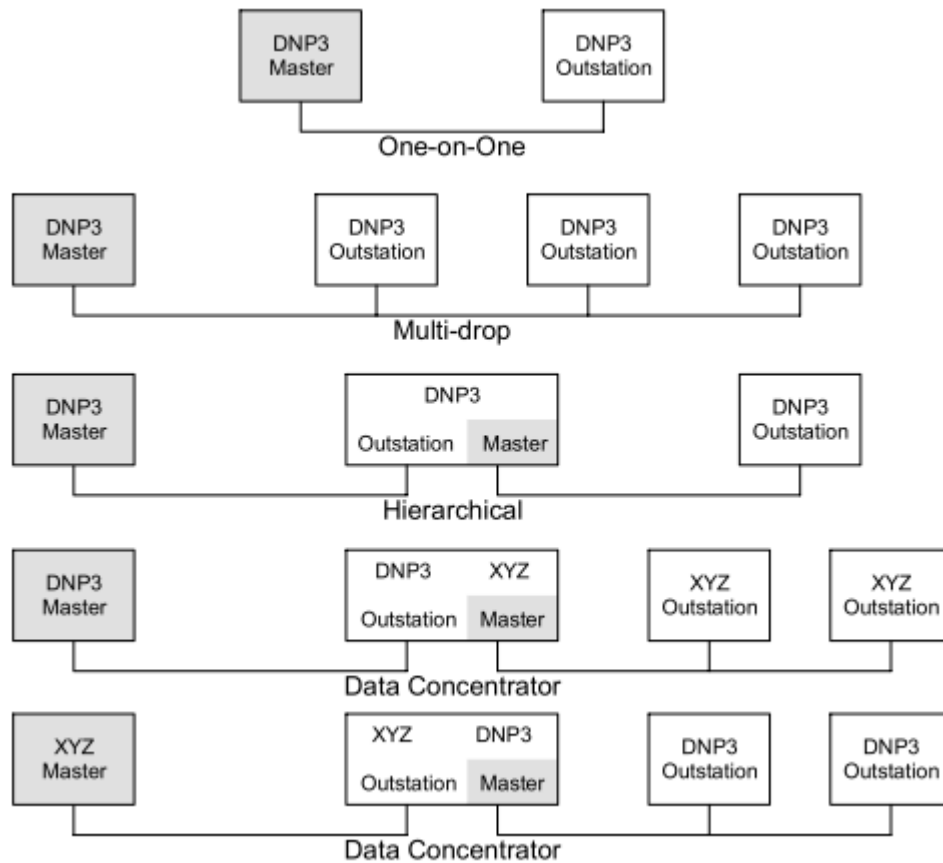
KUVA 4. Näkymä RSCAD:n Runtime-tilasta

## 2.4 DNP3-protokolla

DNP3 (Distributed Network Protocol) on alunperin vuonna 1993 Westronic-yhtiön (nykyisin osa General Electricia) kehittämä tiedonsiirtoprotokolla, joka on IEC 60870-5 -protokollakomponenttien pohjalta kehitetty vastaamaan erityisesti Pohjois-Amerikan vaatimuksia ja tarpeita. Kehittämisen taustalla on ollut saada aikaan avoin ja käyttökelpoinen protokollastandardi, sillä ennen DNP3:n julkaisua ongelmia on tuonut lukuisat eri laitetoimittajien patentoimat protokollat ja kommunikoinnin yhteensovittaminen muiden laitteiden kanssa. Protokollaa käytetään erityisesti sähköverkko- ja vesiyhtiöiden SCADA-valvomoiden ja ala-asemien välisessä kommunikoinnissa. Nykyisin DNP-käyttäjäryhmä omistaa protokollan ja on vastuussa protokollan kehittämisestä. (DNP3 Overview 2002.)

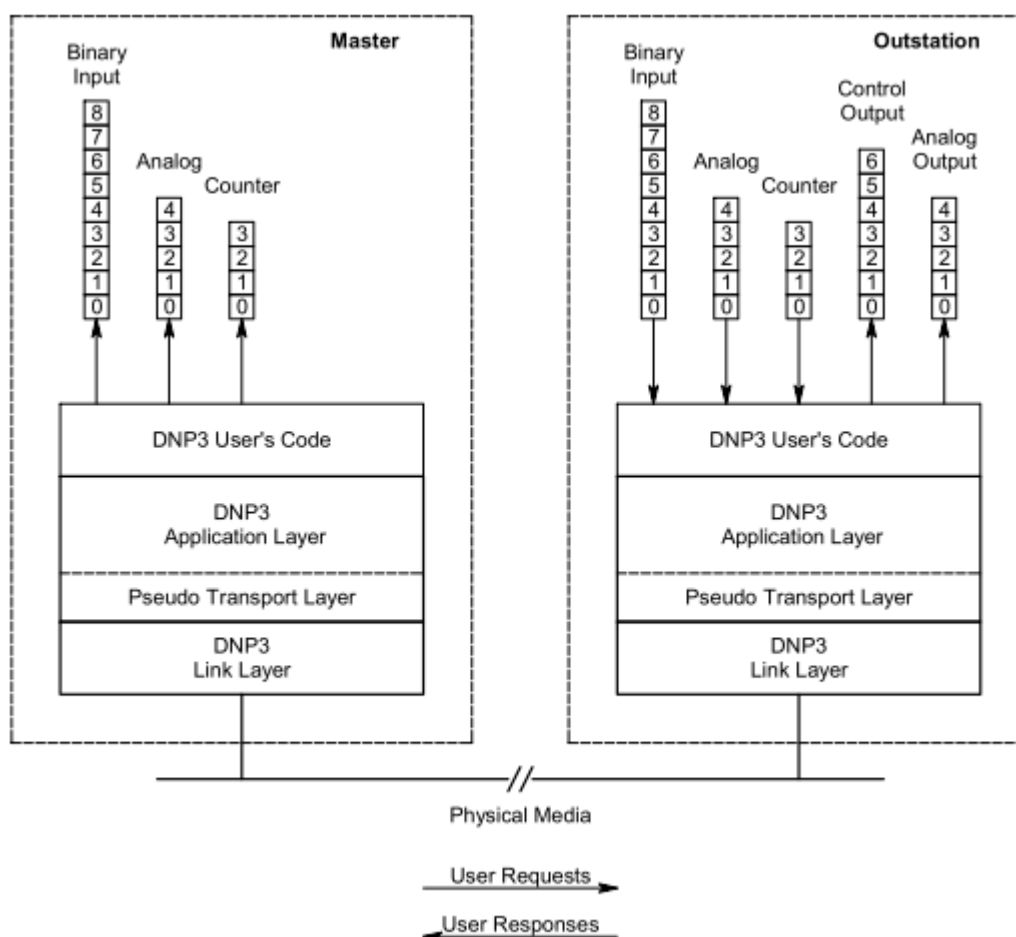
Koska DNP3 on avoin protokolla, se mahdollistaa eri toimittajien laitteiden ja ohjelmien tiedonsiirron yhteensopivuuden. Protokollan ominaisuuksia ovat muun muassa luotettava CRC-virheentarkistusalgoritmi, laatuliput, aikaleimatut signaalit ja ajan synkronisointi. (DNP 2011.)

Protokollan liikennöinti perustuu master–slave-periaatteeseen. Tyypillisesti SCADA-järjestelmä toimii master-yksikkönä, joka kerää tietoa kaikilta siihen kytketyiltä laitteilta sekä lähettää ohjauskomentoja. RTU- ja IED-laitteet toimivat slave-yksikköinä, jotka lähettävät master-yksikölle monitoroimaansa tietoa. Protokolla mahdollistaa liikennöinnin useammalle master-yksikölle. Kuviossa 5 on esitettyä tyypillisiä DNP3-arkkitehtuureita. (DNP 2011.)



KUVIO 5. Tyypillisiä DNP3-arkkitehtuureita (DNP 2011)

Kuviossa 6 on esitetty master- ja slave-yksikön välinen kommunikointiyhteys yksinkertaistettuna. Kuvan oikeassa ylälaudassa on esitettyä numeroitujen neliöiden sisällä slave-yksikön (Outstation) tietokantaan tallennettu data ryhmittäin. Erilaisia dataryhmiä ovat binääriset tulot ja lähdöt, analogiset tulot ja lähdöt sekä laskuriarvot. DNP3-standardin mukaan jokaisessa dataryhmässä yksittäiset signaalit luokitellaan numerojärjestyksessä alkaen nolasta. Samaan tapaan master-yksikön tietokannassa käytetään vastaavia ryhmäkohtaisia numeroita DNP3-osoitteina, niistä signaaleista jotka halutaan siirtää laitteiden välillä. (DNP 2011.)



KUVIO 6. DNP3-kommunikointiyhteys master- ja slave-yksikön välillä (DNP 2011)

OSI-mallin seitsemän kerroksisen ratkaisun sijasta DNP3-protokolla käyttää yksinkertaistettua kolmen kerroksen mallia. Tähän kuuluu sovelluskerros (Application layer), siirtoyhteyskerros (Link layer) ja fyysinen kerros (Physical layer). Lisäksi neljäntenä kerroksena voidaan pitää kuljetuskerrosta (Pseudo-transport layer), joka mahdollistaa sovelluskerroksen sanomien pilkkomisen pienempiin paketteihin alaspäin siirtoyhteyskerrokselle. (DNP3 Overview 2002.)

Fyysisellä kerroksella määritellään tiedonsiirrossa käytettävä fyysisen media. DNP3:n alkuvuosina liikennöinti tapahtui lähinnä hitaiden sarjamuotoisten RS-232- ja RS-485-väylien kautta. Nykyään protokolla on kehittynyt ja se on muunnettu toimimaan pakettikytkentäisten TCP/IP-verkkojen päällä. Tämä mahdollistaa tehokkaan kaistanleveyden hyödyntämisen, minkä vuoksi protokolla soveltuu ympäristöihin, joissa asemat ovat levittäytyneet maantieteellisesti laajalle alueella ja kommunikointi tapahtuu langattomasti, esimerkiksi maksullisessa matkapuhelinverkossa. (DNP3 Overview 2002.)

Siirtoyhteyskerros lähettää ja vastaanottaa paketteja, joita kutsutaan myös kehyksiksi. DNP3-kehys sisältää siirrettävän tiedon lisäksi otsakkeen, jossa määritellään kehyksen koko, hallinnointitiedot sekä lähde- ja kohdeosoitteet. Siirtoyhteyskerroksessa on toteutettu myös CRC-pohjainen siirtovirheiden ja kaksoiskehysten havaitseminen liikennöinnin luotettavuuden lisäämiseksi. (DNP 2011.)

DNP3-tietokannan signaalien päivitys tapahtuu master-yksikön aloitteesta. Master voidaan asetella suorittamaan tietyin väliajoin yleiskysely, jossa päivitetään kaikkien tietokannan signaalien tila. Jokaiselle signaalille voidaan myös asettaa prioriteetti, mikä mahdollistaa esimerkiksi prosessin kannalta kriittisten signaalien kyselyn huomattavasti useammin ja vähemmän tärkeiden signaalien päivityksen hieman harvemmin. Protokolla tukee myös spontaania raportointia, jolloin slave-yksikkö voi lähettää omatoimisesti muuttuneet signaalien arvot masterille. (DNP3 Overview 2002.)

Kuten edellä on käyty läpi, DNP3-protokollalla on lukuisa määrä eri toiminallisuuksia ja protokolla on jaettu neljälle eri toiminnalliselle tasolle. DNP3:n ensimmäinen taso määrittelee yksinkertaisimman toteutuksen master- ja slave-kommunikaation välille, kun taas neljännessä tasossa on määritelty kaikki monimutkaisemmatkin ominaisuudet. Jokaisen DNP3:ta tukevan laitetoimittajan tulee laatia laiteprofiilidokumentti, jossa määritellään tarkasti ennalta määritellyt parametrit, kuten esimerkiksi mikä on laitteen korkein tuettu DNP-taso ja mitä protokollaan liittyviä ominaisuuksia laite tukee. (DNP 2011.)

### 3 SÄÄTÖJÄRJESTELMÄN VIRITTÄMINEN

#### 3.1 PID-algoritmi

PID-säädin (Proportional, Integral, Derivative) on yksi yleisimmistä säätimistä teollisuuden eri säätöprosesseissa. PID-algoritmi on peräisin 1930-luvulta ja nykyisin arviolta 95 % prosessiteollisuuden säätimistä on PID-säätimiä. Säätimiä voidaan käyttää käytännössä kaikissa sovelluksissa, joissa prosessin tulo- ja lähtösuureet on mitattavissa. (Sung & Lee 2009.)

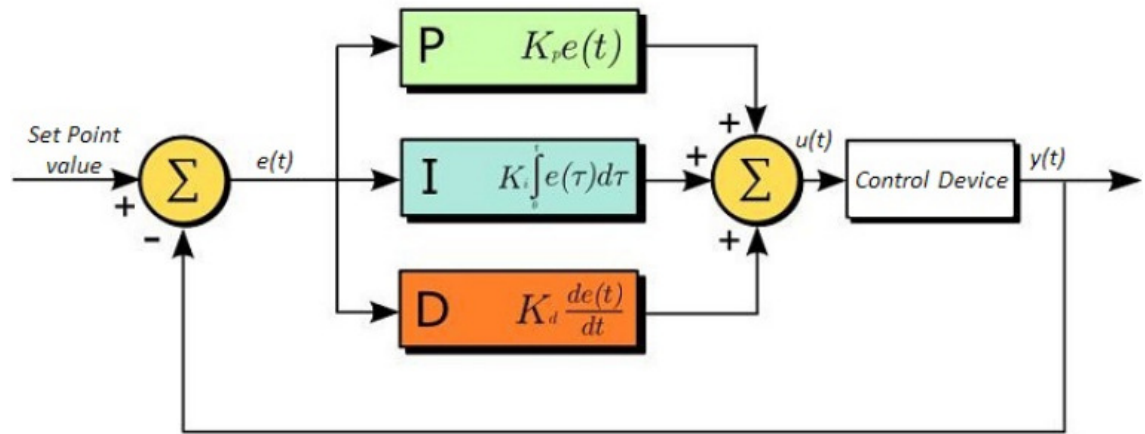
Säätimen etuja on mm. yksinkertaisuus, hyvä suorituskky ja luotettavuus. Vaikka säädin on rakenteeltaan yksinkertainen, se tarjoaa riittävästi monipuolisuutta, mikä mahdollistaa riittävän tarkkuuden monimutkaistenkin prosessien säätöön. Vaikka PID-säädin yleisin säädintyyppi, niin suurta osaa prosessiteollisuuden säätimistä ei ole viritetty optimaalisesti. Oikeilla PID-säätimen parametrien valinnoilla voidaan parantaa prosessin tehokkuutta huomattavasti sekä vähentää toimilaitteille aiheutuvia rasituksia. (Sung & Lee 2009.)

PID-säädin perustuu vahvistimen, integraattorin ja derivaattorin yhdistelmään. Säätimelle tuodaan haluttu asetusarvo ja takaisinkytkentänä systeemistä mitattu arvo. Säädin säätää toimilaitteen (esim. moottori, venttiili yms.) asentoa tai tehoa näiden arvojen perusteella. Erosuure on asetusarvon ja mitta-arvon erotus. Erosuure approksimoi säätövirhettä eli asetusarvon ja säädettävän suureen erotusta. PID-säätimen tuottama ohjaus on summa kolmesta termistä; erosuureeseen suoraan verrannollisesta termistä, erosuureen derivaatista ja erosuureen integraalista.

Takaisinkytketyn PID-säätimen lohkokaavio on esitettynä kuviossa 7. Säätimen lähtö  $u$  on esitettynä kaavassa 4.

$$u = K_P e + K_I \int e dt + K_D \frac{de}{dt} = K_P e + \frac{1}{T_I} \int e dt + T_D \frac{de}{dt}, \quad (4)$$

missä  $e$  on erosuure,  $K_P$  suhdeosan vahvistus,  $K_I$  integrointivahvistus,  $K_D$  derivointivahvistus,  $T_I$  integrointiaika ja  $T_D$  derivointiaika.



KUVIO 7. Takaisinkytketyn PID-säätimen lohkokaavio (Thorlabs 2015)

Suhdeosa (P) muodostaa säätimen ulostulon erosuureeseen suoraan verrannollisen termin. Tällöin P-osan ulostulon itseisarvo on sitä suurempi, mitä kauempana toivotusta säädettävän suureen mittausrvo on. Vahvistus  $K_P$  ilmaisee säätötoimenpiteen voimakkuutta. Jos systeemi sisältää paljon mittaushäiriöitä ei vahvistustermiä voida kasvattaa suureksi, sillä säätimestä tulee epävakaa. Pelkän P-säädön huono puoli on, että käytännössä systeemiin jää aina pysyvän tilan virhettä, eli säädin ei saa korjattua säätövirhettä nolaksi. (Åström & Murray 2008, 293.)

Integroiva osa (I) integroi erosuuretta ajassa. Sen ulostulo riippuu sekä erosuureen suuruudesta että myös sen kestoajasta. Integraalitermi muuttuu aina, kun erosuure ei ole nolla. Integroivan termin vahvistuksena käytetään integrointivahvistusta  $K_I$ . Säätimen integroinnin ansiosta vakioasetusarvolle saatava säätövirhe voidaan kompensoida monissa sovelluksissa, kun P-säädöllä ei siihen pystytä. Integroiva säätö auttaa mm. vakiohäiriöihin ja lineaarisesti muuttuvien asetusarvojen aiheuttamien säätövirheiden eliminoinnissa. Etujen saavuttaminen saattaa kuitenkin edellyttää mittaukselta sopivaa tarkkuutta. Toisaalta suuri integrointitermi lisää värähtelyä ja epästabiilisuutta. Pääsääntöisesti I-termiä ei käytetä yksinään, vaan myös P-termi tarvitaan riittävän stabiiliuden, vaimennuksen ja tehokkuuden saavuttamiseksi. (Åström & Murray 2008, 293.)

Derivoiva osa (D) reagoi erosuureen muutoksiin yrittäen vastustaa niitä. D-termin parametria  $K_D$  kutsutaan derivointivahvistukseksi. Derivoivaa osaa kutsutaan myös ennakoivaksi, koska se pyrkii kompensoimaan poikkeaman jo siinä vaiheessa, kun se vasta on muodostumassa. Sen käyttö helpottaa stabilointia, vaimennuksen ja tehokkuuden saavuttamista. Huonona puolena derivointi vahvistaa erosuureeseen



mahdollisesti sisältyvää korkeataajuisia kohinaa. Tällöin säätimen lähdöstä saattaa tulla niin kohinainen sisäänmeno toimilaitteelle, että toimilaitteen toiminnasta tulee liian rauhaton. Yleensä derivoivaa osaa ei käytetä elleivät muut syyt pakota sen käyttöön. (Åström & Murray 2008, 293.)

Nykyään kaikki sähköisesti toteutetut säätimet tai automaatiojärjestelmien säätölohkot sisältävät vähintäänkin täydellisen PID-algoritmin. Mikäli ei kuitenkaan haluta käyttää täydellistä PID-algoritmia, saadaan siitä PI-algoritmi virittämällä derivointiaikavakio  $T_D$  nolaksi tai vaikkapa P-algoritmi virittämällä edelleen integrointiaikavakio  $T_I$  äärettömäksi. Tällöin voidaan tarpeettomaksi katsottu tai joissain tilanteissa jopa ongelmia tuottava termi jättää säädöstä kokonaan pois. Usein riittävä säätötarkkuus saavutetaan, vaikka kaikkea PID-säätimen potentiaalista ei hyödynnetäkään (Savolainen & Vaitinen 2007, 38).

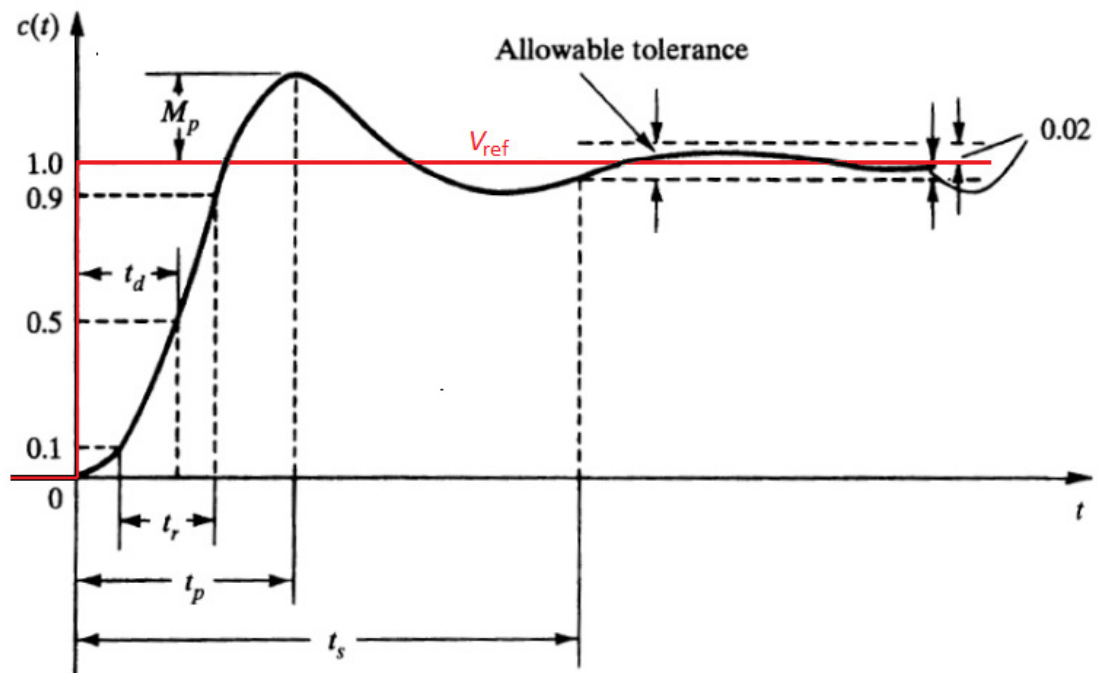
### 3.2 Askelvastetesti

Askelvastetesti on yksi menetelmä todentaa säätöjärjestelmän dynaamista suorituskkyä. Askelvastetestissä tutkitaan järjestelmän määrätyn tulo-lähtö-parin aikakäyttäytymistä. Kokeen edellytyksenä on, että ohjaukseen voidaan tehdä yhtäkkinen askelmainen muutos ja samanaikaisesti sekä ohjaus- että lähtösuure tulee voida mitata ja tallentaa ajan suhteen. (Savolainen & Vaitinen 2007, 38.)

Askelvastekoe aloitetaan kytkemällä järjestelmä käsinohjaukseen, minkä jälkeen se ohjataan johonkin tyypilliseen tasapainotilaansa. Tasapainotilassa järjestelmän tulo- ja lähtösuureet pysyvät vakioina eli ne eivät muutu ajan suhteen. Kun järjestelmän tasapainotila on esim. piirturien avulla todettu, tehdään ohjaukseen askelmainen muutos, jonka tulisi edustaa suuruudeltaan tyypillistä käytännössä esiintyvää ohjausmuutosta. Tämän jälkeen odotetaan, kunnes järjestelmä on asettunut uutta ohjausta vastaavaan tasapainotilaan. Piirturilla nähdään järjestelmän askelvaste eli lähtösuureen käyttäytyminen ajan suhteen, kun ohjauksena on ns. askelfunktio. (Savolainen & Vaitinen 2007, 38.)

Kuviossa 8 on esitettyä askelvastetestin kuvaaja ajan funktiona. Lähtösuure on esitettyä mustalla käyrällä ja ohjaussuure punaisella. Vasteen kuvaajan avulla voidaan

helposti nähdä järjestelmälle tärkeitä ominaisuuksia, kuten asettumisaika tai pysyvä poikkeama. Vasteet ovat hyödyllisiä työkaluja suunnitellessa sähköisien ohjausjärjestelmien säätöä.



KUVIO 8. Askelvastetestin kuvaaja ajanfunktiona

Kun määritellään säädön tavoitteita on huomioitava kuinka tarkasti ja nopeasti säätimen on toimittava. Askelvastetestien yhteydessä saadaan määritettyä hyvyyskriteereiksi kutsuttuja tunnuslukuja, jotka kertovat järjestelmän dynaamisesta suorituskyvystä. Hyvyyskriteereitä ovat mm.

- viive  $t_d$  (delay time).
- nousuaika  $t_r$  (rise time)
- huippuaika  $t_p$  (peak time)
- maksimiylitys  $M_p$  (maximum overshoot) ja
- asettumisaika  $t_s$  (settling time). (Bazoune 2006.)

Viiveellä  $t_d$  tarkoitetaan aikaa, joka vasteelta kuluu saavuttaakseen puoliväli tavoitearvostaan. Nousuaika  $t_r$  on aika, joka vasteelta kuluu noustaessa 10 %:n arvosta 90 %:n arvoon, mutta arvoja voidaan tutkia myös muilta väleiltä. Huippuaika  $t_p$  on aika, jolloin vaste on saavuttanut ensimmäisen huippuarvon ylityksen jälkeen. Ylitys  $M_p$  on vasteen ja loppuarvon erotus ensimmäisen huipun kohdalla. Suuruus ilmaistaan usein prosentteina loppuarvosta. Systemin asettumisaika  $t_s$  on aika, jonka jälkeen askelvaste pysyy kahden prosentin (jossain tapauksissa myös viiden prosentin) sisällä

loppuarvosta. Tarkka matemaattinen lauseke asettumisajalle on hankala, mutta riittävän tarkka arvio saadaan laskemalla värähtelevän vasteen verhoikäyrän asettumisaika. (Bazoune 2006.)

Muita suorituskysyksiä on mm. säätöpoikkeama, eli referenssin ja loppuarvon erotus, sekä kuollut aika. Kuollut aika on aika, joka kuluu tulosuureen muutoksesta ensimmäiseen tilan muutokseen. Viiveettömässä järjestelmässä tulosuureen muutos aiheuttaa välittömän tilan muutoksen, joka tapahtuu järjestelmän aikavakioiden kuvaamalla nopeudella. Viivettä saattaa esiintyä säädettävän järjestelmän lisäksi ohjauksessa ja mittauksessa (Savolainen & Vaitinen 2007, 25).

### 3.3 Säätimen virittäminen

Jotta säädöstä olisi ylipäättään saatavissa jotakin hyötyä, tulee säädön olla hyvin viritetty. Virittämällä tarkoitetaan tässä yhteydessä oikeiden viritysparametrien määrittämistä PID-säätimelle. Yleisesti hyvässä säätöjärjestelmässä ei saa esiintyä liiallista värähtelyä, prosessin tulee pystyä toipumaan mahdollisimman nopeasti häiriön jälkeen ja säätöpoikkeaman eli ohjearvon ja oloarvon erotuksen tulisi olla mahdollisimman pieni. Näillä perusteilla voidaan lähteä virittämään järjestelmää kokeilemalla erisuuria parametriarvoja. (Savolainen & Vaitinen 2007, 41.)

Virheellisesti optimoitu säädin huonontaa sen prosessin laatua, jota se on asetettu valvomaan. Lisäksi mekaaniset käyttö- ja toimilaitteet rasittuvat tarpeettomasti, jos prosessi värähtelee itsekseen (Johansson 2000, 81).

Täydellisen PID-menetelmän virittäminen umpimähkäisesti "yritä ja epäonnistu"-menetelmällä voi olla työläs prosessi. Säättöjärjestelmien virittämiseksi onkin kehitetty erilaisia systemaattisia menetelmiä säädinparametrien määrittämiseksi, esimerkkeinä Ziegler-Nichols- ja Cohen-Coon-menetelmät. Käytännössä on kuitenkin harvinaista, että säätöpiirien viritykseen käytetään teoreettisia menetelmiä. (Savolainen & Vaitinen 2007, 41.)

Järjestelmän voidaan sanoa olevan epästabiili, jos sen vaste kasvaa rajatta. Vastaavasti jos vaste saavuttaa jonkin vakioarvon äärellisessä ajassa, on systeemi stabiili. Kun

SVC:n slope-asetusta kasvatetaan, niin säätimen toiminta nopeutuu, mikä lisää riskiä järjestelmän epästabiilisuuteen. Slope-asetuksen lisäksi SVC:n toimintapiste ja sähköverkon vahvuus eli oikosulkuteho vaikuttaa vasteeseen ja järjestelmän stabiilisuuteen (Nokian Capacitors 2009).

Heikossa verkossa loistehon aiheuttamat jännitemuutokset tapahtuvat nopeasti ja SVC:n vaste on tyypillisesti nopea. Jos verkon oikosulkuteholla on taipumus muuttua SVC:n toimintaympäristössä, tulee säätöjärjestelmä suunnitella adaptiiviseksi, jotta haluttu vasteaika ja säätöpiirin stabiilisuus voidaan säilyttää kaikissa sähköverkon kytkentätilanteissa. (Chun 2000, 2642-2647.)

Säätöpiirin manuaaliselle virittämiselle on olemassa peukalosääntöjä, mutta niiden käyttäminen vaatii prosessin tuntemista. Taulukossa 1 on esitetty PID-parametrien vaikutus askelvastetestien hyvyyskriteereihin, kun parametrin arvoa kasvatetaan.

TAULUKKO 1. PID-parametrien vaikutus hyvyyskriteereihin, kun parametrin arvoa kasvatetaan (Zhong 2006)

| Parametri | Nousuaika    | Ylitys | Asettumisaika | Pysyvän tilan virhe | Stabiilisuus               |
|-----------|--------------|--------|---------------|---------------------|----------------------------|
| $K_p$     | Laskee       | Kasvaa | Pieni muutos  | Laskee              | Huononee                   |
| $K_i$     | Laskee       | Kasvaa | Kasvaa        | Poistuu             | Huononee                   |
| $K_d$     | Pieni muutos | Laskee | Laskee        | Ei vaikutusta       | Paranee jos $K_d$ on pieni |

Viritettävien säätimien lukumäärä, prosessin monimutkaisuus ja järjestelmän dynaamiset ominaisuudet vaikuttavat olennaisesti siihen, mikä viritystapa on kustannustehokkain sovelluskohteessa. Tiedetyt prosessit edellyttävät säätimen nopeaa toimintaa, jolloin ylitystä esiintyy usein askelvastetestissä. Toisaalta jotkut prosessit eivät salli ollenkaan ylityksen esiintymistä, mikä johtaa säätimen hitaampaan viritukseen.

## 4 TESTIAUTOMAATIO

### 4.1 Perusteet

Testiautomaatiossa tietokone ohjelmoidaan matkimaan ihmisen toimintaa. Erillinen automatisoitu ohjelma suorittaa testejä halutulle ohjelmistolle tai järjestelmälle. Monissa mallinnus- ja simulointiohjelmissa on valmiina työkaluja, joita voidaan hyödyntää testiautomaation toteuttamisessa.

Ohjelmien tai ohjausjärjestelmien toiminnan testaaminen voi olla työlästä ja aikaa vievää, jos se tehdään manuaalisesti. Testiautomaatio ei kuitenkaan poista manuaalista testausta, vaan täydentää sitä. Testiautomaatiolla voidaan vähentää riskejä, sillä automatisoidut testit ajetaan aina samalla tavalla. Manuaalisessa testauksessa testejä ei välttämättä ajeta joka kerta samalla tavalla ihmisten virheiden tai unohdusten takia. (Somé 2010.)

Testiautomaatiota suunniteltaessa on otettava huomioon, että automaatiosta saatavat hyödyt ovat suuremmat kuin sen toteuttamiseen käytettävä työpanos. Jos testiä joudutaan toistamaan monta kertaa ja testivaiheet ovat testaajalle raskaita ja monimutkaisia, niin automatisointi voi tuoda huomattavan parannuksen tilanteeseen. Tällöin testiautomaatio tuo säästöjä testauksen toteuttamiseen vähentämällä testien ajamiseen kuluvaa aikaa. Muita testiautomaation hyötyjä on muun muassa helppo toistettavuus, jäljitettävyyys ja parempi laatu. Huonona puolena on, että testityökalut edellyttävät työkalujen ja komentosarjojen ylläpitoa. (Somé 2010.)

Koneiden on vaikea suoriutua joistain tehtävistä, jotka ovat helppoja ihmisille. Esimerkiksi testitulosten analysointi on ihmiselle paljon helpompaa kuin koneelle. Lisäksi koneita voi olla usein vaikea opettaa reagoimaan poikkeaviin tai yllättäviin tilanteisiin testattavan järjestelmän käyttäytymisessä. Testiautomaatiota onkin paras hyödyntää sellaisten testitapausten kanssa, jotka eivät ole alttiita muutoksille.

## 4.2 Testiautomaation työkaluja

Testiautomaatiossa käyttäjä määrittelee erilaisia testitapauksia. Testitapaukset ajetaan testausohjelmaan, joka suorittaa ennalta määritellyn testisekvenssin ja kerää automaattisesti testien tulokset. Alla esitellään tyypillisimpiä testiautomaation työkaluja.

### Nauhoita ja toista

Testiautomaatiota voidaan toteuttaa käyttämällä nauhoita-toista -työkalua, jota käytetään erityisesti graafisten käyttöliittymien yhteydessä. Kun työkalu laitetaan nauhoitus-tilaan, se tallentaa käyttäjän kaikki liikkeet ja toiminnot, kuten hiiren liikkeet, näppäimistön painallukset tai testiparametrien arvojen vaihtamisen. Vastaavasti toistotilassa voidaan toistaa automaattisesti kaikki nauhoitetut toiminnot yhtenä komentosarjana. (Zallar 2000.)

Työkalun etuna on helppokäyttöisyys, sillä käyttäjältä ei vaadita juuri lainkaan ohjelmointitaitoja. Huonona puolena on, että lopputuloksena tulevan komentosarjan ylläpito on vaikeaa, sillä nauhoituksesta syntyvä komentosarja on huonosti strukturoitu. Monesti nauhoitus tulee uusiksi kokonaisuudessaan, vaikka kyseessä olisikin vain pieni muutos.

### Manuaalinen ohjelmointi

Komentosarjojen manuaalinen ohjelmointi on tehokkaampi ja joustavampi työkalu automatisoida testaamista, mutta edellyttää käyttäjältä osaamista ohjelmointisuunnittelusta. Noudattamalla ohjelmointikäytäntöjä saadaan parannettua komentosarjojen ylläpidettävyyttä. Lisäksi ohjelma on helpommin muokattavissa, kun koodin rakenne on hyvin strukturoitu ja dokumentoitu. (Matara 2013.)

### Dataohjattu testaus

Dataohjatussa testauksessa testidata on eriytetty varsinaisesta komentosarjasta. Komentosarjaan tuodaan testiparametrit erillisestä tiedostosta tai tietokannasta ja sijoitetaan ne komentosarjassa käytettäviin muuttujiin. Manuaaliseen ohjelmointiin verrattuna tämä nopeuttaa testaamista, sillä komentosarjoihin ei tarvitse tehdä muutoksia, vaan testiparametrien arvojen vaihtaminen riittää. (Matara 2013.)

**Avain- ja toimitisanaohjattu testaus**

Avain- ja toimitisanapohjainen testaus on edistyneempi muoto dataohjatusta testauksesta. Testidatan lisäksi myös itse komentosarjan suoritussjärjestys ja toiminnallinen logiikka listataan erilliseen taulukkoon, josta sovellus generoi testiskriptin. (Matara 2013.)

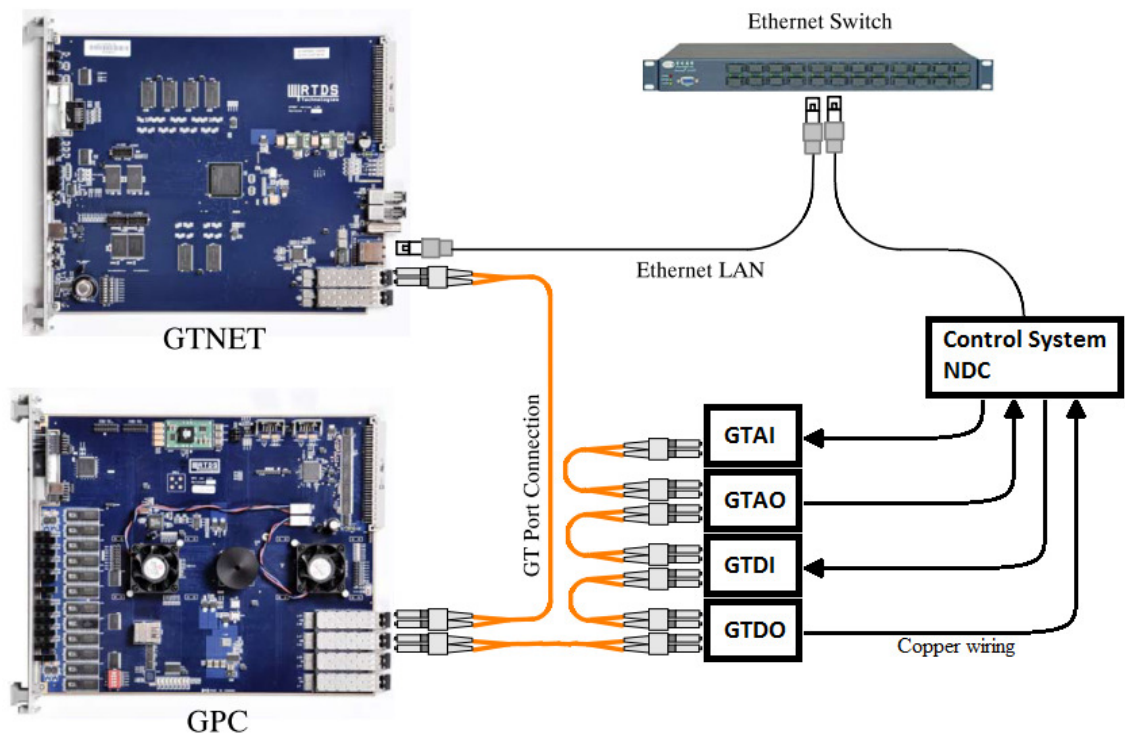
## 5 ASKELVASTETESTIEN AUTOMATISOINTI

### 5.1 Nykytilanne

SVC:n toiminnallisuus testataan kokonaisuudessaan RTDS-testiympäristössä ennen sähköasemalla tehtäviä testejä suurjännitteen kanssa. SVC:n ohjausjärjestelmä sisältää lukuisan määrän eri testattavia kokonaisuuksia. Esimerkiksi ohjausjärjestelmän jännitesäädin viritetään suoriutumaan määritellyistä askelvastevaatimuksista. Asiakkaalta tulevassa laitespesifikaatiossa määritellään askelvastevaatimukset, joista tärkeimmät ovat tyypillisesti nousuaika, ylitys ja asettumisaika.

Kuviossa 9 on esitettyä SVC:n ohjausjärjestelmän testijärjestely yksinkertaistettuna. RTDS:n verkkomallista saadaan GTAO-moduulin kautta tuotua ohjausjärjestelmään mittaussignaaleja, kuten sähköverkon virrat ja jännitteet. Vastaavasti GTAI-moduulin kautta saadaan tuotua RTDS:ään ohjausjärjestelmän säätösuureita tulosten analysointia varten. Analogiasignaaleja saadaan päivitettyä RSCAD:n kuvaajiin, joiden pohjalta voidaan todentaa säädön toiminnallisuus. Lisäksi käytössä on GTDI- ja GTDO-moduuleita binäärisignaaleja varten, joilla saadaan toteutettua kytkinlaitteiden ohjaukset ja tilatiedot RSCAD:n Runtime-sovelluksessa.





Kuvio 9. SVC:n ohjausjärjestelmän RTDS-testijärjestely

PID-säätimen parametrien arvot viritetään kokeellisesti, eli parametreja muutetaan kunnes vaste täyttää vaatimukset. Säätimen toiminta vaihtelee hieman riippuen siitä mikä on sähköverkon tila, SVC:n toimintapiste ja mitkä ovat ohjausjärjestelmän parametrit. Testitapauksien suunnittelussa tulee ottaa huomioon mikä on sähköverkon oikosulkuteho, annetaanko jännitereferenssi induktiiviseen vai kapasitiiviseen suuntaan ja mikä on käytössä oleva slope-asetus. Tämä voi aiheuttaa kymmeniä eri testitapauksia, ja mikäli yhdessäkään tapauksessa ei saavuteta vasteelle asetettuja tavoitteita tulee PID-säätimen parametreja muuttaa ja aloittaa alusta.

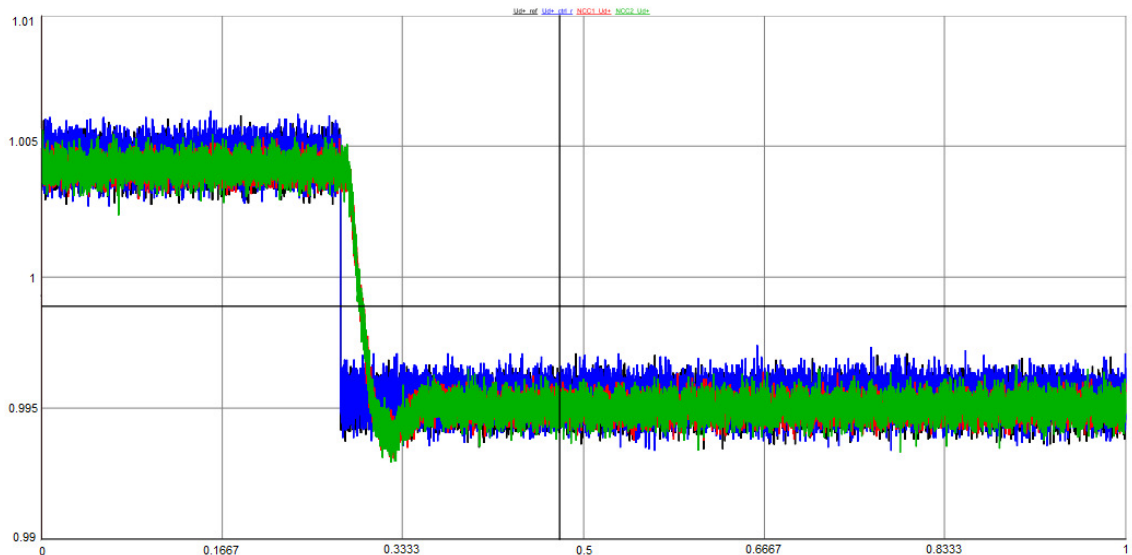
Taulukossa 2 on esitettyä esimerkkinä askelvastetestitapauksia ohjausjärjestelmän ja sähköverkon eri toimintapisteissä.  $V_{ohje1}$ -arvo on ohjausjärjestelmän jänniteohjearvo stabiilissa alkutilanteessa ja  $V_{ohje2}$ -arvo on askelvasteen jänniteohjeen arvo. Asiakkaan vaatimusmäärittely voi lisätä huomattavasti tarkasteltavien testitapauksien määrää.

TAULUKKO 2. Esimerkki askelvastetestitapauksista

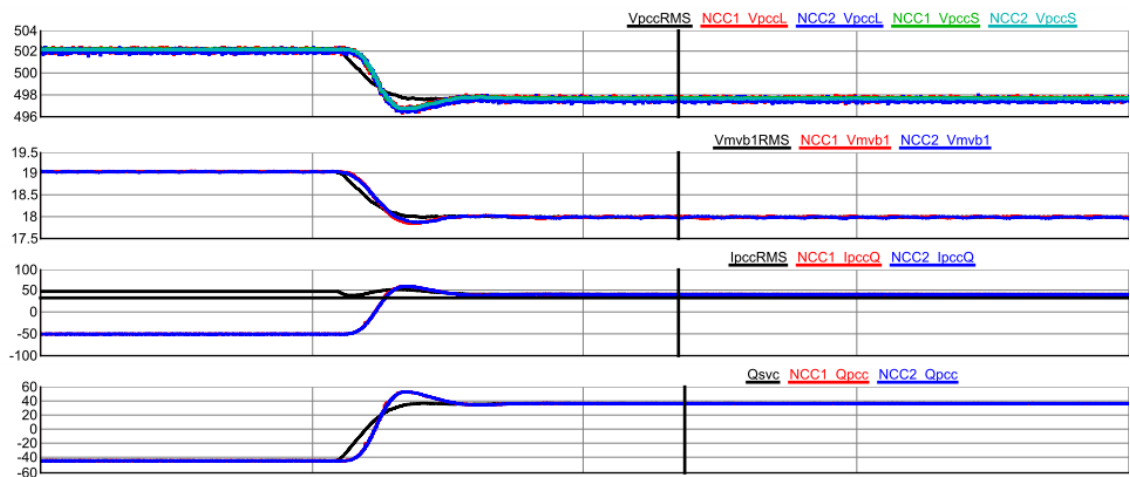
| Testitapaus<br>nro | $V_{\text{verkko}}$<br>[kV] | Oikosulkuteho<br>[MVA] | Slope<br>[%] | $V_{\text{ohje1}}$<br>[pu] | $V_{\text{ohje2}}$<br>[pu] |
|--------------------|-----------------------------|------------------------|--------------|----------------------------|----------------------------|
| 1                  | 400,00                      | 2500,00                | 0,00         | 0,95                       | 0,98                       |
| 2                  | 400,00                      | 2500,00                | 0,00         | 1,05                       | 1,02                       |
| 3                  | 400,00                      | 2500,00                | 5,00         | 0,95                       | 0,98                       |
| 4                  | 400,00                      | 2500,00                | 5,00         | 1,05                       | 1,02                       |
| 5                  | 400,00                      | 2500,00                | 10,00        | 0,95                       | 0,98                       |
| 6                  | 400,00                      | 2500,00                | 10,00        | 1,05                       | 1,02                       |
| 7                  | 405,00                      | 6000,00                | 0,00         | 0,95                       | 0,98                       |
| 8                  | 405,00                      | 6000,00                | 0,00         | 1,05                       | 1,02                       |
| 9                  | 405,00                      | 6000,00                | 5,00         | 0,95                       | 0,98                       |
| 10                 | 405,00                      | 6000,00                | 5,00         | 1,05                       | 1,02                       |
| 11                 | 405,00                      | 6000,00                | 10,00        | 0,95                       | 0,98                       |
| 12                 | 405,00                      | 6000,00                | 10,00        | 1,05                       | 1,02                       |

Normaalissa toiminnassa SVC rampittaa jännitereferenssin halutuun arvoonsa rauhallisesti esimerkiksi 10 sekunnin aikana. Askelvastetestien ajaksi säätöjärjestelmä asetetaan kuitenkin testitilaan, jossa rampitus poistuu ja jännitereferenssi annetaan askelmaisesti. Ennen jokaista testitapausta testaaja asettaa ohjausjärjestelmän ja sähköverkon parametrit manuaalisesti haluttuun arvoon ja antaa askelvasteen jänniteohjearvon muodossa. Testaaja käynnistää manuaalisesti RTDS:n kuvaajien tallennuksen. Tallenteen pituus on tyypillisesti 0,5...3 sekuntia vasteaikavaatimuksista riippuen ja testaajan tulee manuaalisesti ajoittaa ohjausjärjestelmän askelvaste tallenteen alkupäähän.

Kuviossa 10 on esitettyä esimerkkinä RSCAD:n tulostama sekunnin mittainen askelvastekuvaaja, jossa näkyy sinisellä jänniteohje ja vihreällä säädettävä jännite. Kuviossa 11 on puolestaan esitettyä samasta testitapauksesta sähköverkon liityntäpisteen jännitteen, SVC-kiskon jännitteen ja virran sekä SVC:n tuottaman tehon käyttäytyminen askelmaisen jännitemuutoksen seurauksena. Kuvaajista nähdään muun muassa, että aluksi SVC tuottaa 40 MVAria kapasitiivista loistehoa ja jänniteohjearvon muutoksen jälkeen 40 MVAria induktiivista loistehoa. Samalla sähköverkon jännite tippuu 502 kV:sta 498 kV:iin.



KUVIO 10. Esimerkki RSCAD:n tulostamasta askelvastekuvaajasta



KUVIO 11. Kuvaaja SVC:n ja sähköverkon suureiden käyttäytymisestä askelvasteessa

RSCAD:ssä on toiminto, jolla saadaan tallennettua valitut kuvaajat PDF- ja COMTRADE-formaateissa. Tallennuksen yhteydessä annetaan kuvaajille nimi tunnistusta varten. Jokaisen askelvastetestitapauksen jälkeen RTDS:stä saadut kuvaajat analysoidaan ja lasketaan manuaalisesti askelvasteen tunnusluvut. Riippuen kuvaajien määrästä ja tallenteen pituudesta aikaa kuluu kuvaajien piirtymisen odotteluun ja tallentamiseen, joka tehdään manuaalisesti. Lisäksi parametrien asetteleminen käsin kunkin testitapauksen välillä on aikaa vievää.

## 5.2 Automatisoinnin toteutus

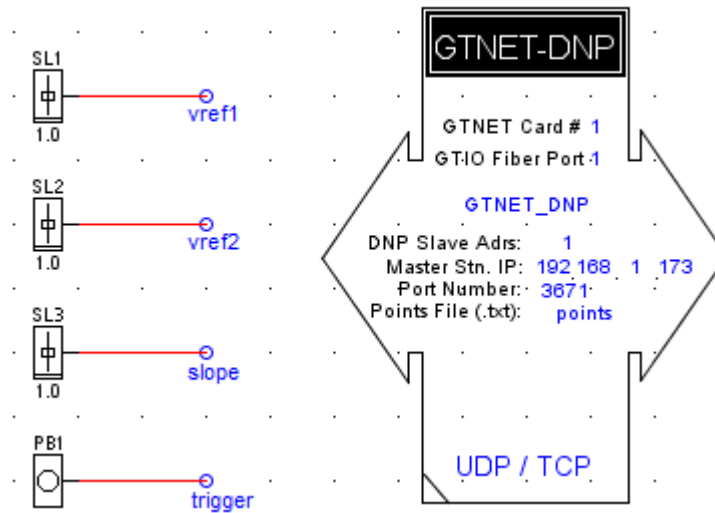
RSCAD tukee C-kieleen pohjautuvia komentosarjoja, joiden avulla voidaan rakentaa monimutkaisiakin testisekvenssejä. Lisäksi RSCAD:ssä on komentosarjojen nauhoitustoiminto, jolla voidaan helposti toteuttaa yleisimmin käytettäviä toimintoja RSCAD:n Runtime-tilassa. Komentosarjojen avulla voidaan toteuttaa automaattisesti parametrien muuttaminen jokaista testitapausta varten. Tässä työssä automatisointi on toteutettu näitä työkaluja käyttäen.

Komentosarja tallentaa automaattisesti kaikista testitapauksista kuvaajat PDF- ja COMTRADE-formaateissa myöhempää analysointia varten. Myös testin aikana käytetty simulaatiomalli voidaan tallennetaan käyttäjä valitsemaan hakemistoon.

### 5.2.1 DNP3-protokollan käyttöönotto

Työssä on luotu RTDS:n ja NDC-ohjausjärjestelmän välille kommunikointirajapinta, jotta halutut signaalit saadaan kulkemaan järjestelmien välillä. RTDS-räkkiin kytketään GTNET-kortti, jonka avulla voidaan muodostaa yhteyksiä ulkoisiin laitteisiin eri kommunikointiprotokollien kautta. GTNET-kortti tukee DNP3-protokollaa, mutta kortti voi toimia ainoastaan slave-yksikkönä, jolloin NDC-ohjausjärjestelmä asetetaan toimimaan master-yksikkönä.

Kuviossa 12 on esitettyä RSCAD-malliin uutena lisättävät komponentit automatisointiin liittyen. SVC-simulaatiomallissa voidaan jo ennestään muuttaa sähköverkon jännitelähteen ja oikosulkutehon arvoja, mutta jännitereferenssit ja slope-asetus ovat SVC-ohjausjärjestelmän sisäisiä parametreja. Näistä signaaleista tehdään RSCAD-malliin kolme liukusäätimellä ohjattavaa muuttujaa. DNP3-kommunikaatiota varten komponentti, johon määritetään mm. master-yksikön IP-osoite, porttinumero ja DNP3-pistelistan tiedostonimi. Lisäksi tarvitaan painonapilla toimiva tahdistussignaali, jonka avulla ajoitetaan askelvaste sopivalle kohtaa tallennettavaa kuvaajaa.



KUVIO 12. RSCAD-malliin lisättävät komponentit

Signaalit lisätään GTNET-kortin DNP3-pistelistaan ja SVC:n ohjausjärjestelmän CMI-kortille määritellään vastaavat pisteet DNP3-konfiguraatitiedostoon. Kuviossa 13 on esitettyä GTNET-kortin DNP3-signaalikonfiguraatio, johon määritetään RSCAD-mallissa käytettävät signaalien nimet, DNP-osoitteet ja analogiatulo-signaalien deadband-arvot.

```
# DNP3 signal point mapping
#      DNP_address      user_label      bitmap_name      bitmap_bit_num
BI:    0                BIT_STS        trigger          0

#      DNP_address      variable_name  deadband%
AI:    0                vref1        1%
AI:    1                vref2        1%
AI:    2                slope        1%
```

KUVIO 13. GTNET-kortin DNP3-signaalit

Slave-yksikkönä toimiva GTNET-kortti ei tue spontaaneja viestejä valmistajan ilmoittaman DNP3-laiteprofiilin (liite 1) mukaan, jolloin muuttuneiden signaalien arvot eivät päivitty automaattisesti masterina toimivalle ohjausjärjestelmälle. Tämän vuoksi ohjausjärjestelmän CMI-kortin kiertokyselyaika (polling time) tulee konfiguroida mahdollisimman lyhyeksi, esimerkiksi 100 ms:iin, jotta muuttuneet signaalit saadaan riittävän nopeasti päivitettyä ohjausjärjestelmään.

### 5.2.2 RSCAD-komentosarjat

Työtä varten on luotu Excel-taulukko, jonka avulla komentosarja voidaan luoda automaattisesti. Kuviossa 14 on esitetty näkymä Excel-taulukosta. Aluksi käyttäjän tulee määrittää yleiset RTDS-malliin liittyvät lähtöarvot ja jokaisen testitapauksen parametrit Excel-taulukkoon. Komentosarja luo jokaisesta parametrasta kolumnittain taulukkotyyppiset muuttujat, jotka esimerkkitapauksessa sisältävät 12 arvoa. Excel-taulukon yhteyteen on luotu VBA-makro, joka kirjoittaa automaattisesti askelvastetesteissä käytettävän komentosarjan C-kielelle RSCAD:n ymmärtämään muotoon.

|                     |                              |   |
|---------------------|------------------------------|---|
| Script save folder: | C:\temp                      | Script is created here                            |
| Model folder:       | C:\RTDS_USER\fileman\project | Folder where RSCAD model is located               |
| Model name:         | SVC                          | Name of model used in RSCAD                       |
| Save model folder:  | C:\SVC_step_tests\models     | Models are saved here                             |
| Save record folder: | C:\SVC_step_tests\records    | Records are saved here                            |
| Plot numbers:       | Plot1, Plot2, Plot3          | Name of plots to be saved. Use ',' as a separator |

| Case name | Vsource [kV] | SCL [MVA] | Slope [%] | VrefStart [pu] | VrefEnd [pu] | TSC discharge |
|-----------|--------------|-----------|-----------|----------------|--------------|---------------|
| 1         | 500,00       | 4000,00   | 0,00      | 0,95           | 1,05         | 1             |
| 2         | 500,00       | 4000,00   | 0,00      | 1,05           | 0,95         | 0             |
| 3         | 500,00       | 4000,00   | 5,00      | 0,95           | 1,05         | 1             |
| 4         | 500,00       | 4000,00   | 5,00      | 1,05           | 0,95         | 0             |
| 5         | 500,00       | 4000,00   | 10,00     | 0,95           | 1,05         | 1             |
| 6         | 500,00       | 4000,00   | 10,00     | 1,05           | 0,95         | 0             |
| 7         | 500,00       | 8000,00   | 0,00      | 0,95           | 1,05         | 1             |
| 8         | 500,00       | 8000,00   | 0,00      | 1,05           | 0,95         | 0             |
| 9         | 500,00       | 8000,00   | 5,00      | 0,95           | 1,05         | 1             |
| 10        | 500,00       | 8000,00   | 5,00      | 1,05           | 0,95         | 0             |
| 11        | 500,00       | 8000,00   | 10,00     | 0,95           | 1,05         | 1             |
| 12        | 500,00       | 8000,00   | 10,00     | 1,05           | 0,95         | 0             |

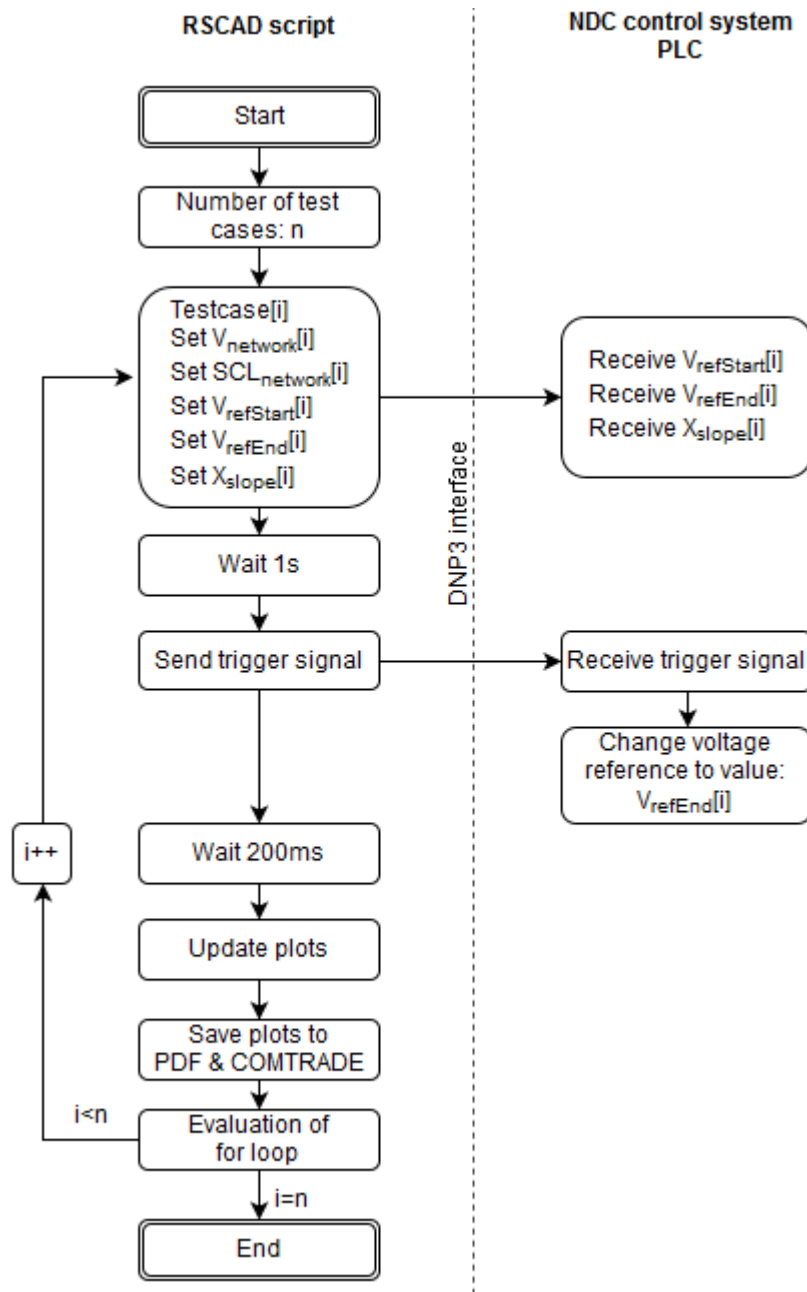
Create Script

Create Script

KUVIO 14. Näkymä komentosarjan luomisessa käytettävästä Excel-taulukosta

Riippuen askelvasteen suuruudesta kapasitiiviselle tehoalueelle säätöjärjestelmä voi reagoida vasteeseen kytkemällä TSC:n päällä, jolloin simulaatiomallin TSC-kondensaattoreihin jää jännitevaraus. Jännitevaraus saadaan poistettua verkkomallissa tekemällä muutaman sekunnin pituinen maasulku TSC-venttiilille, jolloin jäännösvaraus purkautuu nopeasti kondensaattorien sisäisten resistanssien kautta.

Kuvassa 15 on esitetty lohkokaavio komentosarjan toiminnasta. Toiminta perustuu for-silmukan käyttämiseen, missä suoritetaan sama testisekvenssi jokaisen testitapauksen parametrien arvoilla.



KUVIO 15. Lohkokaavio komentosarjan toiminnasta

Ennen jokaista testitapausta asetetaan verkkomallin ja ohjausjärjestelmän parametrit taulukon mukaisiin alkuarvoihin, minkä jälkeen odotetaan hetki, jonka aikana sähköverkko palautuu muutoksen jälkeen stabiiliin tilaan. Ohjausjärjestelmä tekee askelmaisen jänniteohjearvon muutoksen komentosarjan merkkisignaalista. DNP3-kommunikoinnista aiheutuvan signaalinsiirron viiveen vuoksi odotetaan 200 ms ennen kuin päivitetään RSCAD:n kuvaajat, jolloin askelvaste saadaan näkyviin aina kuvaajan alkupäähän. Tämän jälkeen toistetaan komentosarja seuraavan testitapausten parametreilla, kunnes kaikista tapauksista on saatu kuvaajat tallennettua.

Kuviossa 16 on esitettynä näkymä C-kielisestä komentosarjasta, jossa näkyy alkua for-silmukasta. SetSlider-komennolla saadaan asetettua RSCAD-mallin liukusäätimet haluttuun arvoon. Kuvaajien päivitys on lukittuna parametrien asettamisen aikana ja lukitus vapautetaan kun on ohjausjärjestelmä muuttaa jänniteohjearvoa.

```
for (i=0;i<12;i++)
{
    loop_counter++;
    fprintf(stdmsg,"Running simulation Case Number %d\n",loop_counter);

    MasterPlotLockState = 1;
    SetSlider "Subsystem #1 : CTLS : Inputs : SL1" = vref1[i];
    SetSlider "Subsystem #1 : CTLS : Inputs : SL1" = vref2[i];
    SetSlider "Subsystem #1 : CTLS : Inputs : SL2" = slope[i];
    SetSlider "Subsystem #1 : CTLS : Inputs : SL3" = SCL[i];
    SetSlider "Subsystem #1 : CTLS : Inputs : SL4" = Vnetwork[i];
    SetDial "Subsystem #1 : CTLS : Inputs : SCL" = SCLDial[i];
    SUSPEND 1.0;
```

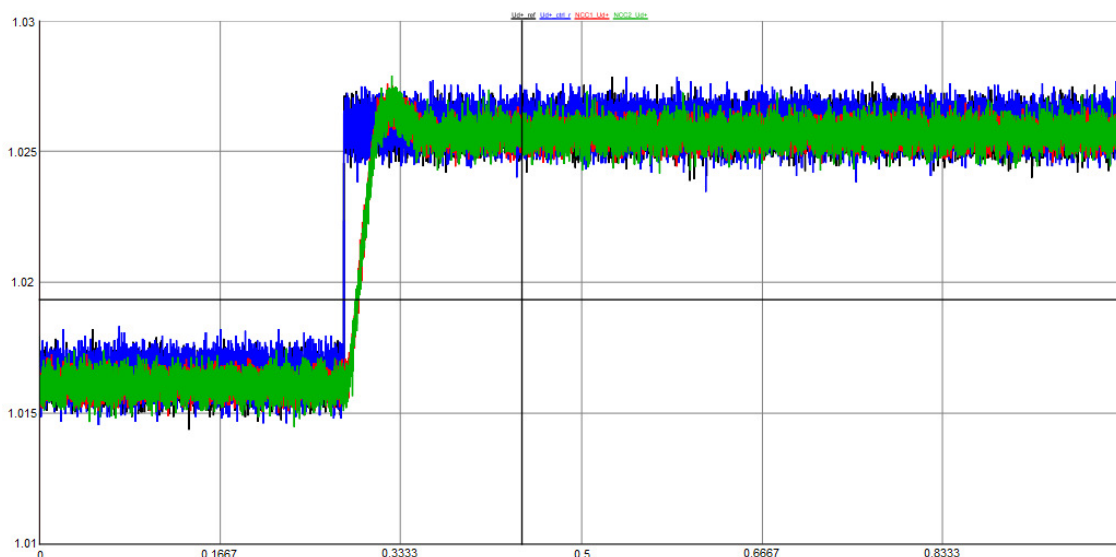
KUVIO 16. Näkymä RSCAD:n komentosarjasta

RSCAD-komentosarjan lisäksi myös ohjausjärjestelmän PLC-ohjelmaan tulee luoda yksinkertainen testiohjelma, joka asetetaan päälle askelvastetestin yhteydessä. PLC-ohjelmassa kirjoitetaan jännitereferenssi  $V_{\text{refEnd}}$ -muuttujan arvoon, kun komentosarjan lähettämä tahdistussignaali on vastaanotettu.

### 5.3 Automatisoinnin tulokset

Kuvassa 17 on esitettynä RSCAD:n tulostama sekunnin mittainen kuvaaja askelvasteesta, jossa jännitteet on ilmoitettu pu-arvoina. Sininen käyrä on jänniteohjearvo ja vihreä käyrä on säädettävä suure eli sähköverkon liityntapisteen kiskojännite. Vaikka mitatuissa signaaleissa on hieman kohinaa, voidaan kuvaajan perusteella määrittää vasteen nousuaika, ylitys ja asettumisaika.





KUVIO 17. RSCAD:n askelvastekuvaaja

Askelvastekuvaajat näyttävät yhdenmukaisilta, sillä komentosarjan aikaansaama jännitteohjearvon muutos osuu aina samalle kohdalle kuvaajissa. Manuaalisesti tehtynä vasteen saaminen kuvaajaan näkyviin käsin vaatii testaajalta hyvää reaktiokykyä, ja ilman automaatiota sama testitapaus pitää usein toistaa monta kertaa ennen kuin jännitemuutos saadaan ajoitettua sopivalle kohdalle.

Komentosarja toteuttaa ennalta määriteltyä sekvenssiä ja suorittaa jokaisen testitapauksen täsmälleen annettujen parametrien perusteella. Tämän vuoksi inhimillisten virheiden määrä vähenee ja testaajan helppo tarkastella automaation toimintaa.

Excel-taulukko ja sen kautta generoitu komentosarja on suunniteltu helppokäyttöiseksi, jolloin käyttäjä pystyy tekemään nopeasti muokkauksia kuhunkin käyttötilanteeseen sopivaksi. Testien tekeminen uudelleen tai niihin palaaminen jälkikäteen helpottuu, sillä testisekvenssit ja käytetyt parametrit ovat nopeasti palautettavissa.

Askelvastetestien automatisoinnilla on tarkoitus tehostaa testeihin käytettävää aikaa. Automaatiolla vähennetään manuaalisen parametrien muuttamiseen, kuvaajien piirtymiseen ja tallentumiseen kuluva aikaa. Testisekvenssi siirtyy seuraavaan vaiheeseen automaattisesti heti kun siihen on valmius, jolloin testaajan reagointiin kuluva viive poistuu ja ylimääräinen odottamiseen käytettävä aika vähenee. Koska muutettavia parametrejä ja yksittäisiä testitapauksia voi olla kymmenittäin, korostuu automatisoinnin tuomat hyödyt.

## 6 YHTEENVETO

Lopputyön tavoitteena oli kehittää menetelmä SVC:n ohjausjärjestelmän testaamisen automatisoimiseksi. Askelvastetestien automatisoinnilla on suuri merkitys tehdastestauksen prosessin ja työajan tehostamisen näkökulmasta, vaikka kyseessä onkin vain yksi testattava kokonaisuus. SVC:n dynaamisen suorituskyvyn todentaminen on kuitenkin merkittävä osa tehdastestejä.

Testiautomaatiossa tarvitaan aina ihmisten tekemää työtä. Testiautomaatio ei poista manuaalisen testauksen tarvetta, vaan auttaa testaajia keskittymään muihin monimutkaisempiin testeihin. Testauksen huolellinen suunnittelu on tärkeää, jotta voidaan määritellä, mitkä testit kannattaa lopulta automatisoida, sillä testiautomaation ohjelmointi ja komentosarjojen testaaminenkin vaatii työtunteja. Automaatio tuo testaukseen lisäarvoa erityisesti silloin, kun sitä hyödynnetään usein toistettaviin testitapauksiin.

Askelvastetestien tekeminen manuaalisesti on yksitoikkoista ja aikaa vievää, ja vie testaajien resursseja heidän ydinsaaamisestaan. Alustavien tulosten perusteella voidaan olettaa, että testaaminen komentosarjojen avulla on huomattavasti tehokkaampaa ajankäytöllisesti manuaalitestaukseen verrattuna. Tämän johdosta työllä on suuret edellytykset tulla käyttöön vakituksena työkaluna osana yrityksen projektien tehdastestejä.

Lopputyön tekemisen ajankohtaan ei ollut saatavilla toimitettavan projektin ohjausjärjestelmää testilaboratoriossa, joten askelvastetestejä ei voitu tehdä täydellisellä laitteistolla. Ohjausjärjestelmän ja RTDS:n välisen DNP3-kommunikaation sekä RSCAD:n komentosarjojen testaaminen on kuitenkin antanut lupaavia tuloksia kehitetyn menetelmän toimivuudesta. Komentosarjojen laajamittaisempi testaaminen on suunnitelmissa seuraavan projektin ohjausjärjestelmän tehdastestauksen yhteydessä.

Helpon muokattavuuden ansiosta komentosarjoja voidaan kehittää jatkossa projektien tarpeiden mukaisesti, mikä tekee työkalusta pitkäikäisen. Tulevaisuudessa ohjausjärjestelmän testaamisen automatisointia voidaan laajentaa koskemaan myös muitakin testikokonaisuuksia kuin askelvastetestejä.

Vaikka automatisoinnilla on saatu toteutettua askelvastekuvaajien luominen automaattisesti, testaajan tulee silti analysoida tulokset manuaalisesti kuvaajien perusteella. Jatkokehityskohteena onkin askelvastetestien tulosten automaattinen analysointi ja sen toteutus. Viemällä askelvaste COMTRADE-muodossa johonkin kolmannen osapuolen ohjelmaan (esim. MATLABiin), voitaisiin automaattisesti määrittää ohjelmallisesti askelvasteen tunnusluvut.

## LÄHTEET

Bazoune, A. 2006. ME 413 System Dynamics & Control. Course material. Luettu 15.11.2015. [http://opencourseware.kfupm.edu.sa/colleges/ces/me/me413/files%5C2-Chapters\\_10\\_5\\_Transient\\_Response\\_Specifications.pdf](http://opencourseware.kfupm.edu.sa/colleges/ces/me/me413/files%5C2-Chapters_10_5_Transient_Response_Specifications.pdf)

Chun, L. 2000. Investigation of voltage regulation stability of static synchronous compensator in power system. IEEE: Power Engineering Society Winter Meeting (Vol. 4).

Distributed Network Protocol. 2001. DNP Users Group. Luettu 14.11.2015. <https://www.dnp.org/default.aspx/>

Johansson, E. 2000. Sääätötekniikka 2000. Iisalmi: IS-VET Oy.

Kangasala SVC. 2008. Projektikuvahakemisto. Alstom Grid Oy.

Kangasala SVC ohjausjärjestelmän spesifikaatio. 2009. Tampere: Nokian Capacitors Oy.

Kundur, P. 1994. Power system stability and control. United States of America: McGraw-Hill Inc.

Matara, M. & Vuori, M. & Jääskeläinen A. 2013. Ohjelmistojen testaus. Tampereen teknillinen yliopisto. Luettu 25.2.2016. [http://www.cs.tut.fi/~testaus/s2013/luennot/TIE-21200\\_2013.pdf](http://www.cs.tut.fi/~testaus/s2013/luennot/TIE-21200_2013.pdf)

RTDS Technologies Inc. 2015. Luettu 14.11.2015. <https://www.rtds.com/>

Savolainen, J. & Vaittinen, R. 2007. Sääätötekniikan perusteita. Helsinki: Hakapaino Oy.

Somé, S. 2010. Introduction to Automated Testing. University of Ottawa. Luettu 25.2.2016. <http://www.site.uottawa.ca/%7Essome/Cours/CSI5118/IntroAutoTesting.pdf>

Sung, S., Lee, J. & Lee, I. 2009. Process Identification and PID Control. USA: John Wiley & Sons Inc.

Thorlabs Inc. 2015. PID basics. Luettu 14.11.2015. <http://www.thorlabs.de/tutorials.cfm?tabID=5dfca308-d07e-46c9-baa0-4defc5c40c3e>

Triangle MicroWorks Inc. 2002. DNP3 Overview. Luettu 14.11.2015. <https://www.trianglemicroworks.com/>

Zallar, K. 2000. Automated Software Testing - A Perspective. Methods & Tools. Luettu 25.2.2016. <http://www.methodsandtools.com/archive/archive.php?id=33>


Zhong, J. 2006. PID Controller Tuning: A Short Tutorial. USA: Purdue University

Åström, K. & Murray, R. 2008. Feedback Systems: An Introduction for Scientists and Engineers. USA: Princeton University Press

# LIITTEET

## Liite 1. GTNET DNP Device Profile

1 (5)

|   |   |
|---|---|
|  | <b>GTNET-DNP</b><br>Version 1.7, 2009-01-09 |
|---|---|

### Device Profile

Vendor Name: **RTDS Technologies**  
 Device Name: **GTNET-DNP**  
 Highest DNP Level Supported: \_\_\_\_\_ Device Function:

|                               |   |
|-------------------------------|---|
| For Requests: <b>Level 3</b>  | <input type="checkbox"/> Master   |
| For Responses: <b>Level 3</b> | <input checked="" type="checkbox"/> Slave (LAN/WAN only, single master) |

Notes:

For static (non-change-event) object requests, request qualifier codes 07 and 08 (limited quantity), and 17 and 28 (index) are supported. Static object requests sent with qualifiers 07, or 08, will be responded with qualifiers 00 or 01.

Binary input event change event capacity: 2048  
 Analog input event change event capacity: 501

Time format is UTC

Applies to GTNET-DNP firmware versions  
     1.5 for original 2U chassis based GTNET  
     1.6 for current backplane GTNET card

|   |   |
|---|---|
| Maximum Data Link Frame Size (octets):  | Maximum Application Fragment Size (octets):   |
| Transmitted: 292  | Transmitted: 2048   |
| Received: 292   | Received: 2048  |
| Maximum Data Link Re-tries:   | Maximum Application Layer Re-tries:   |
| <input checked="" type="checkbox"/> None  | <input checked="" type="checkbox"/> None  |
| <input type="checkbox"/> Fixed  | <input type="checkbox"/> Configurable   |
| <input type="checkbox"/> Configurable   |   |
| Requires Data Link Layer Confirmation:  |   |
| <input checked="" type="checkbox"/> Never   |   |
| <input type="checkbox"/> Always   |   |
| <input type="checkbox"/> Sometimes  |   |
| <input type="checkbox"/> Configurable as:   |   |
| Requires Application Layer Confirmation:  |   |
| <input type="checkbox"/> Never  |   |
| <input type="checkbox"/> Always   |   |
| <input checked="" type="checkbox"/> When reporting Event Data                       |   |
| <input type="checkbox"/> When sending multi-fragment responses (Slave devices only) |   |
| <input type="checkbox"/> Sometimes  |   |
| <input type="checkbox"/> Configurable   |   |
| Timeouts while waiting for:   |   |
| Data Link Confirm:  | <input checked="" type="checkbox"/> None <input type="checkbox"/> Fixed at _____ <input type="checkbox"/> Variable <input type="checkbox"/> Configurable. |
| Complete Appl. Fragment:  | <input checked="" type="checkbox"/> None <input type="checkbox"/> Fixed at _____ <input type="checkbox"/> Variable <input type="checkbox"/> Configurable  |
| Application Confirm:  | <input type="checkbox"/> None <input checked="" type="checkbox"/> Fixed at 10 s <input type="checkbox"/> Variable <input type="checkbox"/> Configurable.  |
| Complete Appl. Response:  | <input checked="" type="checkbox"/> None <input type="checkbox"/> Fixed at _____ <input type="checkbox"/> Variable <input type="checkbox"/> Configurable  |

Others: **Select/Operate Arm Timeout 10 s**

RTDS Technologies Inc., 300-137 Innovation Drive, Winnipeg, Manitoba CANADA R3T 6B6  
 (telephone): +204 989 9700 (fax): +204 452 4303 (email): [rtids@rtids.com](mailto:rtids@rtids.com)



GTNET-DNP  
Version 1.7, 2009-01-09

#### Sends/Executes Control Operations:

|                         |   |  |   |                                       |
|-------------------------|---|--|---|---------------------------------------|
| WRITE Binary Outputs    | <input checked="" type="checkbox"/> Never | <input type="checkbox"/> Always            | <input type="checkbox"/> Sometimes            | <input type="checkbox"/> Configurable |
| SELECT/OPERATE          | <input type="checkbox"/> Never            | <input checked="" type="checkbox"/> Always | <input type="checkbox"/> Sometimes            | <input type="checkbox"/> Configurable |
| DIRECT OPERATE          | <input type="checkbox"/> Never            | <input checked="" type="checkbox"/> Always | <input type="checkbox"/> Sometimes            | <input type="checkbox"/> Configurable |
| DIRECT OPERATE – NO ACK | <input type="checkbox"/> Never            | <input checked="" type="checkbox"/> Always | <input type="checkbox"/> Sometimes            | <input type="checkbox"/> Configurable |
| Count > 1               | <input checked="" type="checkbox"/> Never | <input type="checkbox"/> Always            | <input type="checkbox"/> Sometimes            | <input type="checkbox"/> Configurable |
| Pulse On                | <input type="checkbox"/> Never            | <input type="checkbox"/> Always            | <input checked="" type="checkbox"/> Sometimes | <input type="checkbox"/> Configurable |
| Pulse Off               | <input checked="" type="checkbox"/> Never | <input type="checkbox"/> Always            | <input type="checkbox"/> Sometimes            | <input type="checkbox"/> Configurable |
| Latch On                | <input type="checkbox"/> Never            | <input checked="" type="checkbox"/> Always | <input type="checkbox"/> Sometimes            | <input type="checkbox"/> Configurable |
| Latch Off               | <input type="checkbox"/> Never            | <input checked="" type="checkbox"/> Always | <input type="checkbox"/> Sometimes            | <input type="checkbox"/> Configurable |
| Queue                   | <input checked="" type="checkbox"/> Never | <input type="checkbox"/> Always            | <input type="checkbox"/> Sometimes            | <input type="checkbox"/> Configurable |
| Clear Queue             | <input checked="" type="checkbox"/> Never | <input type="checkbox"/> Always            | <input type="checkbox"/> Sometimes            | <input type="checkbox"/> Configurable |

#### Notes:

Pulse On range is 200 – 10000 msec. Maximum 16 simultaneous pulse operations

Reports Binary Input Change Events when no specific variation requested:

- ☐ Never
- ☒ Only time-tagged
- ☐ Only non-time-tagged
- ☐ Configurable to send one or the other

Reports time-tagged Binary Input Change Events when no specific variation requested:

- ☐ Never
- ☐ Binary Input Change With Time
- ☒ Binary Input Change With Relative Time
- ☐ Configurable

#### Sends Unsolicited Responses:

- ☒ Never
- ☐ Configurable
- ☐ Only certain objects
- ☐ Sometimes (attach explanation)
- ☐ ENABLE/DISABLE UNSOLICITED Function codes supported

#### Sends Static Data in Unsolicited Responses:

- ☒ Never
- ☐ When Device Restarts
- ☐ When Status Flags Change

No other options are permitted.

#### Default Counter Object/Variation:

- ☒ No Counters Reported
- ☐ Configurable
- ☐ Default Object
- ☐ Default Variation:
- ☐ Point-by-point list attached

#### Counters Roll Over at:

- ☒ No Counters Reported
- ☐ Configurable (attach explanation)
- ☐ 16 Bits
- ☐ 32 Bits
- ☐ Other Value: \_\_\_\_\_
- ☐ Point-by-point list attached

#### Sends Multi-Fragment Responses:

- ☒ Yes
- ☐ No
- ☐ Configurable

#### Sequential File Transfer Support:

|                               |                              |  |
|-------------------------------|------------------------------|--|
| Append File Mode              | <input type="checkbox"/> Yes | <input checked="" type="checkbox"/> No |
| Custom Status Code Strings    | <input type="checkbox"/> Yes | <input checked="" type="checkbox"/> No |
| Permissions Field             | <input type="checkbox"/> Yes | <input checked="" type="checkbox"/> No |
| File Events Assigned to Class | <input type="checkbox"/> Yes | <input checked="" type="checkbox"/> No |
| File Events Send Immediately  | <input type="checkbox"/> Yes | <input checked="" type="checkbox"/> No |
| Multiple Blocks in a Fragment | <input type="checkbox"/> Yes | <input checked="" type="checkbox"/> No |
| Max Number of Files Open      | 0                            |  |

RTDS Technologies Inc., 300-137 Innovation Drive, Winnipeg, Manitoba CANADA R3T 6B6  
(telephone): +204 989 9700 (fax): +204 452 4303 (email): [rtids@rtids.com](mailto:rtids@rtids.com)



GTNET-DNP  
Version 1.7, 2009-01-09

### Implementation Table

| OBJECT        |                             |   | REQUEST<br>(Library will parse)                                  |  | RESPONSE<br>(Library will respond with) |  |
|---------------|-----------------------------|---|--|--|---|--|
| Object Number | Variation Number            | Description   | Function Codes (dec)   | Qualifier Codes (hex)  | Function Codes (dec)                    | Qualifier Codes (hex)                              |
| 1             | 0                           | Binary Input – Any Variation                          | 1 (read)   | 00, 01 (start-stop)<br>06 (no range, or all)<br>07, 08 (limited qty)<br>17, 27, 28 (index) |   |  |
| 1             | 1<br>(default – see note 1) | Binary Input  | 1 (read)   | 00, 01 (start-stop)<br>06 (no range, or all)<br>07, 08 (limited qty)<br>17, 27, 28 (index) | 129 (response)                          | 00, 01 (start-stop)<br>17, 28 (index – see note 2) |
| 1             | 2                           | Binary Input with Status                              | 1 (read)   | 00, 01 (start-stop)<br>06 (no range, or all)<br>07, 08 (limited qty)<br>17, 27, 28 (index) | 129 (response)                          | 00, 01 (start-stop)<br>17, 28 (index – see note 2) |
| 2             | 0                           | Binary Input Change – Any Variation                   | 1 (read)   | 06 (no range, or all)<br>07, 08 (limited qty)  |   |  |
| 2             | 1                           | Binary Input Change without Time                      | 1 (read)   | 06 (no range, or all)<br>07, 08 (limited qty)  | 129 (response)                          | 17, 28 (index)                                     |
| 2             | 2                           | Binary Input Change with Time                         | 1 (read)   | 06 (no range, or all)<br>07, 08 (limited qty)  | 129 (response)                          | 17, 28 (index)                                     |
| 2             | 3<br>(default – see note 1) | Binary Input Change with Relative Time                | 1 (read)   | 06 (no range, or all)<br>07, 08 (limited qty)  | 129 (response)                          | 17, 28 (index)                                     |
| 10            | 0                           | Binary Output – Any Variation                         | 1 (read)   | 00, 01 (start-stop)<br>06 (no range, or all)<br>07, 08 (limited qty)<br>17, 27, 28 (index) |   |  |
| 10            | 1                           | Binary Output   | 1 (read)   | 00, 01 (start-stop)<br>06 (no range, or all)<br>07, 08 (limited qty)<br>17, 27, 28 (index) | 129 (response)                          | 00, 01 (start-stop)<br>17, 28 (index – see note 2) |
| 10            | 2<br>(default – see note 1) | Binary Output Status                                  | 1 (read)   | 00, 01 (start-stop)<br>06 (no range, or all)<br>07, 08 (limited qty)<br>17, 27, 28 (index) | 129 (response)                          | 00, 01 (start-stop)<br>17, 28 (index – see note 2) |
| 12            | 1                           | Control Relay Output Block                            | 3 (select)<br>4 (operate)<br>5 (direct op)<br>6 (dir. op, noack) | 17, 28 (index)   | 129 (response)                          | echo of request                                    |
| 30            | 0                           | Analog Input - Any Variation                          | 1 (read)   | 00, 01 (start-stop)<br>06 (no range, or all)<br>07, 08 (limited qty)<br>17, 27, 28 (index) |   |  |
| 30            | 5<br>(default – see note 1) | Short floating point Analog Input                     | 1 (read)   | 00, 01 (start-stop)<br>06 (no range, or all)<br>07, 08 (limited qty)<br>17, 27, 28 (index) | 129 (response)                          | 00, 01 (start-stop)<br>17, 28 (index – see note 2) |
| 32            | 0                           | Analog Change Event – Any Variation                   | 1 (read)   | 06 (no range, or all)<br>07, 08 (limited qty)  |   |  |
| 32            | 5<br>(default – see note 1) | Short floating point Analog Change Event without Time | 1 (read)   | 06 (no range, or all)<br>07, 08 (limited qty)  | 129 (response)                          | 17, 28 (index)                                     |
| 40            | 0                           | Analog Output Status                                  | 1 (read)   | 00, 01 (start-stop)<br>06 (no range, or all)<br>07, 08 (limited qty)<br>17, 27, 28 (index) |   |  |
| 40            | 3<br>(default – see note 1) | Short floating point Analog Output Status             | 1 (read)   | 00, 01 (start-stop)<br>06 (no range, or all)<br>07, 08 (limited qty)<br>17, 27, 28 (index) | 129 (response)                          | 00, 01 (start-stop)<br>17, 28 (index – see note 2) |
| 41            | 3                           | Short floating point Analog Output Block              | 3 (select)<br>4 (operate)<br>5 (direct op)<br>6 (dir. op, noack) | 17, 28 (index)<br>27 (index)   | 129 (response)                          | echo of request                                    |

RTDS Technologies Inc., 300-137 Innovation Drive, Winnipeg, Manitoba CANADA R3T 6B6  
(telephone): +204 989 9700 (fax): +204 452 4303 (email): [rtids@rtids.com](mailto:rtids@rtids.com)



GTNET-DNP  
Version 1.7, 2009-01-09

| OBJECT                                      |                             |                                  | REQUEST<br>(Library will parse) |   | RESPONSE<br>(Library will respond with) |                            |
|---|-----------------------------|----------------------------------|---------------------------------|---|---|----------------------------|
| Object Number                               | Variation Number            | Description                      | Function Codes (dec)            | Qualifier Codes (hex)                         | Function Codes (dec)                    | Qualifier Codes (hex)      |
| 50  | 0                           | Time and Date                    |                                 |   |   |                            |
| 50  | 1<br>(default – see note 1) | Time and Date                    | 1 (read)                        | 07 (limited qty = 1)                          | 129 (response)                          | 07 (limited qty) (qty = 1) |
|   |                             |                                  | 2 (write)                       | 07 (limited qty = 1)                          |   |                            |
| 50  | 3                           | Time and Date Last Recorded Time | 1 (write)                       | 07 (limited qty)                              |   |                            |
| 51  | 1                           | Time and Date CTO                |                                 |   | 129 (response)                          | 07 (limited qty) (qty = 1) |
| 51  | 2                           | Unsynchronized Time and Date CTO |                                 |   | 129 (response)                          | 07 (limited qty) (qty = 1) |
| 52  | 1                           | Time Delay Coarse                |                                 |   | 129 (response)                          | 07 (limited qty) (qty = 1) |
| 52  | 2                           | Time Delay Fine                  |                                 |   | 129 (response)                          | 07 (limited qty) (qty = 1) |
| 60  | 0                           | Not Defined                      |                                 |   |   |                            |
| 60  | 1                           | Class 0 Data                     | 1 (read)                        | 06 (no range, or all)                         |   |                            |
| 60  | 2                           | Class 1 Data                     | 1 (read)                        | 06 (no range, or all)<br>07, 08 (limited qty) |   |                            |
| 60  | 3                           | Class 2 Data                     | 1 (read)                        | 06 (no range, or all)<br>07, 08 (limited qty) |   |                            |
| 60  | 4                           | Class 3 Data                     | 1 (read)                        | 06 (no range, or all)<br>07, 08 (limited qty) |   |                            |
| 80  | 1                           | Internal Indications             | 1 (read)                        | 00, 01 (start-stop)                           | 129 (response)                          | 00, 01 (start-stop)        |
|   |                             |                                  | 2 (write)<br>(see note 3)       | 00 (start-stop)<br>index=7                    |   |                            |
| No Object (function code only) – see note 4 |                             |                                  | 13 (cold restart)               |   |   |                            |
| No Object (function code only) – see note 4 |                             |                                  | 14 (warm restart)               |   |   |                            |
| No Object (function code only)              |                             |                                  | 23 (delay meas.)                |   |   |                            |
| No Object (function code only)              |                             |                                  | 24 (record current time)        |   |   |                            |

Note 1: A Default variation refers to the variation responded when variation 0 is requested and/or in class 0, 1, 2, or 3 scans.

Note 2: For static (non-change-event) objects, qualifiers 17 or 28 are only responded when a request is sent with qualifiers 17 or 28, respectively. Otherwise, static object requests sent with qualifiers 00, 01, 06, 07, or 08, will be responded with qualifiers 00 or 01. (For change-event objects, qualifiers 17 or 28 are always responded.)

Note 3: Writes of Internal Indications are only supported for index 7 (Restart IIN1-7).

Note 4: Does not restart GTNET





GTNET-DNP  
Version 1.7, 2009-01-09

### Point Assignment/Information

The implementation uses a polling mechanism to receive messages from a single master station. The polling rate by the DNP master should be less than 5 Hz.

| Type   | Quantity | Scan rate | Change Event Buffer size | DNP Object Numbers/Names                          | Class Assignment | Notes  |
|--|----------|-----------|--------------------------|---|------------------|--|
| Binary simulation status (breaker position, etc)   | 1024     | 1000 Hz   | 2048                     | 1-Binary Input<br>2-Binary Input Change           | 1                |  |
| Binary simulation controls (breaker controls, etc) | 512      | 1000 Hz   | N/A                      | 10-Binary Output<br>12-Control Relay Output Block | None             |  |
| Analog simulation status (V, I, power, etc)        | 500      | 4 Hz      | 501                      | 30-Analog Input<br>32-Analog Input Change         | 2                | Deadband is user configurable. User defines a deadband as a fixed % of current value |
| Analog simulation controls (analog set-points)     | 100      | 4 Hz      | N/A                      | 40-Analog Output<br>41-Analog Output Block        | None             |  |